Graph Reasoning for Question Answering with Triplet Retrieval

Shiyang Li[1]*, **Yifan Gao**[2], **Haoming Jiang**[2], **Qingyu Yin**[2], **Zheng Li**[2], **Xifeng Yan**[1]
Chao Zhang[3], **Bing Yin**[2]
[1]University of California, Santa Barbara
[2]Amazon Inc.
[3]Georgia Institute of Technology
{shiyangli,xyan}@cs.ucsb.edu
{yifangao,jhaoming, qingyy, amzzhe, alexbyin}@amazon.com
chaozhang@gatech.edu

Abstract

Answering complex questions often requires reasoning over knowledge graphs (KGs). State-of-the-art methods often utilize entities in questions to retrieve local subgraphs, which are then fed into KG encoder, e.g. graph neural networks (GNNs), to model their local structures and integrated into language models for question answering. However, this paradigm constrains retrieved knowledge in local subgraphs and discards more diverse triplets buried in KGs that are disconnected but useful for question answering. In this paper, we propose a simple yet effective method to first retrieve the most relevant triplets from KGs and then rerank them, which are then concatenated with questions to be fed into language models. Extensive results on both CommonsenseQA and OpenbookQA datasets show that our method can outperform state-of-the-art up to 4.6% absolute accuracy.

1 Introduction

Answering complex questions is a challenging task since it often requires world knowledge and reasoning capability of underlying models (Li et al., 2019; Yasunaga et al., 2021; Zhang et al., 2022). Pre-trained language models, e.g. BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), have shown promising results by fine-tuning on downstream question answering tasks. However, world knowledge and reasoning of these models are learned from unstructured data, e.g. Wikipedia text, and are still limited (Li et al., 2019; Petroni et al., 2019).

On the other hand, there exist large-scale knowledge graphs (KGs), e.g. Freebase (Bollacker et al., 2008) and ConceptNet (Speer et al., 2016), capturing world knowledge explicitly by triples to record relations between entities (Zhang et al., 2022). However, how to effectively integrate KGs into language models for question answering is still an open research problem. Li et al. (2019); Yu et al. (2020); Ye et al. (2019); Zhang et al. (2019); Moiseev et al. (2022) focus on utilizing KGs to construct distant supervision signals for continuous pre-training, however, KGs are often dynamic in practice and it is often hard to edit knowledge in models without further training, limiting their usage. Bosselut et al. (2019); Wang et al. (2020) linearize reasoning paths in KGs and train language models on them to generate novel knowledge triplet during inference. However, KGs are discarded after training and language models can hallucinate false world knowledge (Ji et al., 2022).

Lin et al. (2019); Feng et al. (2020); Yasunaga et al. (2021); Zhang et al. (2022); Jiang et al. (2022); Wang et al. (2022) instead first recognize entities in questions and link them to KGs to retrieve subgraphs as additional input besides questions. However, this paradigm constrains retrieved knowledge in local subgraphs and discards more diverse triplets buried in KGs that are disconnected but useful for question answering. In addition, they require extra KG encoders with parameters trained from scratch besides standard language models, limiting model performance when training data is limited.

Recently, there have been growing interests to convert KG as a list of passages represented as natural languages. Oguz et al. (2022); Ma et al. (2021a) convert KG triples into texts and combine these KG-converted texts with heterogeneous resources, e.g. tables and unstructured Wikipedia documents, as passages and achieved state-of-the-art performance for open-domain question answering. Li et al. (2022) follow this line of work and utilize KG and unstructured documents for knowledge-grounded dialogue generation. Zha et al. (2021) linearize reasoning paths in KGs for relation prediction and achieve state-of-the-art performance.

In this paper, we propose to conduct reasoning over KGs for question answering with triplet

*Work was done during internship at Amazon.

Findings of the Association for Computational Linguistics: ACL 2023, pages 3366–3375

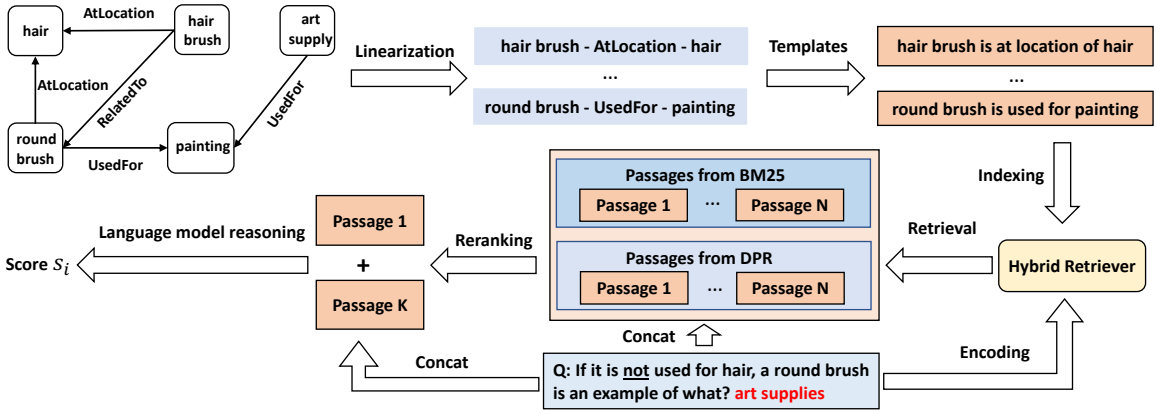

Figure 1: Overview of our framework. The exemplar KG is from Yasunaga et al. (2021).

retrieval following Oguz et al. (2022); Ma et al. (2021a); Li et al. (2022); Zha et al. (2021). The overall pipeline of proposed method is shown in Figure 1. Specifically, we first linearize KG into triplets and convert them into passages by templates and directly retrieve the most relevant ones by questions with both sparse BM25 (Robertson et al., 1994) and Dense Passage Retriever (DPR) (Karpukhin et al., 2020). We then rerank these passages by pre-trained cross-encoders (Reimers and Gurevych, 2019). Finally, most relevant passages and questions are linearly concatenated and fed into pre-trained language model for question answering. This paradigm has several advantages compared to recent state-of-the-art (Yasunaga et al., 2021; Zhang et al., 2022; Jiang et al., 2022; Wang et al., 2022): (1) it is simple yet effective and can outperform state-of-the-art complicated question answering systems up to 4.6% absolute accuracy (2) it does not need extra KG encoders trained from scratch, e.g. GNNs (Scarselli et al., 2009; Veličković et al., 2018), and simply fuses knowledge passages and questions for question answering by standard language models.

2 Methodology

2.1 Problem setup

We focus on multi-choice question answering (MCQA) tasks requiring model reasoning capability. Specifically, for each instance in a MCQA dataset, we have a question q and a candidate choice set $C = \{c_1, c_2, \cdots, c_n\}$. We also assume that we have access to a knowledge graph G, which provides possibly relevant knowledge to answer each question. Given an example (q, C) and a knowledge graph G, we aim to find the correct answer $c^\star \in C$.

2.2 Knowledge graphs as passage corpus

Knowledge graph G can be represented as a list of triplets P. For each triplet $p \in P$, we convert it into natural language passage d by templates so that relevant knowledge can be better retrieved. Specifically, for each triplet $p \in P$, it has a head entity h, a relation r and a tail entity t. We map r into r_p and d is formed by linearly concatenating <h, r_p, t>. For example, if we have a triplet <"hair brush", "AtLocation", "hair">, we map "AtLocation" into "is at location of" and form passage d as "hair brush is at location of hair". Consequently, knowledge graph G is converted into passage corpus D.

2.3 Hybrid passage retrieval

We retrieve passages from corpus D for an MCQA example (q, C) with hybrid (i.e., both sparse and dense) retrievers since they are complementary (Karpukhin et al., 2020; Ma et al., 2021b). For sparse retriever, we utilize BM25 to index D and (q, c_i) to retrieve N passages from it. For dense retriever, we use DPR due to its strong performance in open domain question answering (Karpukhin et al., 2020). DPR embeds queries with question encoder and passages with passage encoder into low dimensional dense vectors, and retrieval can be efficiently done through FAISS library (Johnson et al., 2021) on GPUs. Similar to sparse retriever, we utilize (q, c_i) to retrieve N passages from corpus D for DPR. The total number of passages returned by BM25 and DPR is $2N$.

2.4 Reranking

We further rerank $2N$ passages retrieved by hybrid retriever with pre-trained cross-encoder (Reimers

and Gurevych, 2019). Specifically, for each passage p among retrieved $2N$ passages, we concatenate query (q, c_i) and passage p as the input of pre-trained cross-encoder. For each input, it will output a scalar value between 0 to 1. These scalar values are then used as reranking scores for $2N$ passages.

2.5 Language model reasoning

After reranking, we choose top K passages P_K and concatenate them along with question q and choice c_i, which we cast as input of pre-trained language model (PLM). For input $<q, c_i, P_K>$, PLM will output contextual representation vector $\mathbf{h_i}$, which is then fed into a multi-layer perceptron (MLP) to output a scalar value s_i,

$$\mathbf{h_i} = \text{PLM}(q, c_i, P_K), \quad (1)$$

$$s_i = \text{MLP}(\mathbf{h_i}), \quad (2)$$

where s_i is the prediction score of choice c_i to be correct. During training, we calculate score s_i for each choice $c_i \in C$ and normalize them with softmax function. After that, models are trained to maximize scores of correct choices with standard cross-entropy loss between predictions and ground truth labels. During inference, we calculate score s_i for each choice $c_i \in C$ and select the one with the highest score as the predicted answer of question q.

3 Experiments

3.1 Experimental setups

We evaluate our method on two question answering datasets requiring model reasoning capability.

(1) **CommonsenseQA** (Talmor et al., 2019) is a 5-way multi-choice question answering dataset that requires common sense reasoning. Since its test set is not publicly available, we report in-house split (Lin et al., 2019) for comparisons with baselines.

(2) **OpenbookQA** is a 4-way multi-choice question answering dataset requiring multi-hop reasoning on scientific knowledge (Mihaylov et al., 2018). It has 4957/500/500 questions for training/development/test set split, respectively, and we report results on its test set.

Knowledge graph. For knowledge graph, we utilize ConceptNet (Speer et al., 2016), which is a multi-relational and multi-lingual general knowledge graph storing world common sense knowledge. We first extract English triplets, clean them following (Yasunaga et al., 2021) and convert them into natural language sentences as described in section 2.2, resulting in 2,180,391 passages after data prepossessing. We defer details of relation mapping into Appendix A.

Retrievers. For sparse retriever, we utilize implementation of BM25 from `rank-bm25` python package [1] with default hyperparameters. For dense retriever, we utilize official pre-trained checkpoint [2] from DPR github repository [3].

Reranking. We rerank retrieved passages from BM25 and DPR using pre-trained cross-encoder checkpoint from `sentence-transformers` package [4]. Specifically, for CommonsenseQA dataset, we use pre-trained checkpoint `cross-encoder/ms-marco-MiniLM-L-12-v2` on MS MARCO dataset for passage ranking [5] while for OpenbookQA dataset, we use pre-trained checkpoint `cross-encoder/stsb-roberta-large` on semantic textual similarity task (Cer et al., 2017).

Language model reasoning. Following Yasunaga et al. (2021); Zhang et al. (2022); Jiang et al. (2022); Wang et al. (2022), we utilize RoBERTa-large (Liu et al., 2019) to reason over passages and questions although our framework is model-agnostic. Specifically, question q, choice c_i and passage list P_K are linearly concatenated with special tokens among them and fed into models detailed in section 2.5 to predict choice score.

We defer more implementation and training details of our method into Appendix B.

Baselines. We experiment to compare our method to various baselines with extra KG-encoders, including RN (Santoro et al., 2017), GconAttn (Wang et al., 2018), RGCN (Schlichtkrull et al., 2017), KagNet (Lin et al., 2019), MHGRN (Feng et al., 2020), QA-GNN (Yasunaga et al., 2021), GreaseLM (Zhang et al., 2022), SAFE (Jiang et al., 2022) and GSC (Wang et al., 2022). For these baselines, RoBERTa-large (Liu et al., 2019) is used for both CommonsenseQA and OpenbookQA. We also include RoBERTa-large fine-tuning only baseline without

[1] https://github.com/dorianbrown/rank_bm25

[2] https://dl.fbaipublicfiles.com/dpr/checkpoint/retriver/multiset/hf_bert_base.cp

[3] https://github.com/facebookresearch/DPR

[4] https://www.sbert.net/docs/pretrained_cross-encoders.html

[5] https://github.com/microsoft/MSMARCO-Passage-Ranking

Methods	CommonsenseQA		OpenbookQA
	IHdev-Acc	IHTest-Acc	
RoBERTa-large (w/o KG)	73.07 (±0.45)†	68.69 (±0.56)†	64.80 (±2.37)†
RGCN (Schlichtkrull et al., 2017)	72.69 (±0.19)†	68.41 (±0.66)†	62.45 (±1.57)†
GconAttn (Wang et al., 2018)	72.61(±0.39)†	68.59 (±0.96)†	64.75 (±1.48)†
KagNet (Lin et al., 2019)	73.47 (±0.22)†	69.01 (±0.76)†	-
RN (Santoro et al., 2017)	74.57 (±0.91)†	69.08 (±0.21)†	65.20 (±1.18)†
MHGRN (Feng et al., 2020)	74.45 (±0.10)†	71.11 (±0.81)†	66.85 (±1.19)†
QA-GNN (Yasunaga et al., 2021)	76.54 (±0.21)†	73.41 (±0.92)†	67.80 (±2.75)†
GreaseLM (Zhang et al., 2022)	78.5 (±0.5)⋆	74.2 (±0.4)⋆	66.9¶
SAFE (Jiang et al., 2022)	-	74.03⋆	69.20⋆
GSC (Wang et al., 2022)	<u>79.11</u> (±0.22)†	<u>74.48</u> (±0.41)†	<u>70.33</u> (±0.81)†
Ours	**79.80** (± 0.25)	**74.97** (± 0.56)	**74.93** (± 0.90)

Table 1: Performance comparison in accuracy (%) on both CommonsenseQA and OpenBookQA datasets. We report the average results over three random seeds along with standard deviation on IHdev and IHTest (Lin et al., 2019) for CommonsenseQA dataset and test set performance on OpenbookQA dataset. Best results are bold and second best ones are underlined. †: results from Wang et al. (2022). ⋆: results from their original papers. ¶: results from Yasunaga et al. (2022).

access to extra KG to show the effectiveness of our method. For all experiments in this work, we utilize accuracy (%) as our evaluation metric.

3.2 Main results

As shown in Table 1, our method can consistently outperform state-of-the-art on both CommonsenseQA and OpenbookQA datasets. For CommonsenseQA, our method' test performance can outperform fine-tuned RoBERa-large without KG 6.28% absolute accuracy and outperform best baseline GSC with KG 0.49%. On the smaller dataset OpenbookQA, our method' improvement is larger and can outperform fine-tuned RoBERa-large without KG 10.13% absolute accuracy and outperform best baseline GSC with KG 4.60% accuracy. Note that a key difference between our method and RoBERa-large without KG baseline is that our model also takes additional retrieved passages as input without introducing any extra parameters but can outperform various state-of-the-art methods with extra KG encoders. These consistent results indicate that our simple method can integrate knowledge into language models effectively.

3.3 Ablation study

We further ablate our method by removing BM25 retriever, DPR retriever and reranking module. Note that when we remove reranking module and we use the average score of BM25 and DPR if the same passage is retrieved; otherwise, following (Ma et al., 2021b), if a passage p from BM25 is not in the top N of DPR, we use the lowest score in DPR' top N, and vice versa. Ablation results on IHTest (Lin et al., 2019) of CommonsenseQA and test set of OpenbookQA are shown in Table 2.

Method	CommonsenseQA	OpenbookQA
GSC (best baseline)	74.48(±0.41)	70.33(±0.81)
Ours	**74.97** (± 0.56)	**74.93** (± 0.90)
- BM25	71.45 (±0.17)	<u>72.47</u> (± 0.57)
- DPR	<u>74.81</u> (±1.35)	71.73 (± 0.25)
- Reranking	73.87 (±0.85)	70.93 (± 1.16)

Table 2: Results of removing BM25, DPR and Reranking module on both CommonsenseQA and OpenbookQA dataset.

When removing BM25, results on CommonsenseQA and OpenbookQA drop significantly up to 3.52%. When removing DPR, the result on CommonsenseQA drops slightly while the result on OpenbookQA drops 3.20%. These results show that BM25 and DPR are complementary, aligning with Ma et al. (2021b). Similarly, when further removing reranking module, model performance on CommonsenseQA drops 1.10% and on OpenbookQA drops 4.00% accuracy. These consistent results show the effectiveness of reranking passages retrieved from hybrid retriever. In addition, the <u>second best</u> model shown in our ablation study can still achieve strong performance and outperform best baseline GSC. These consistent results indicate that our model benefits from the combination of sparse and dense retrievers, and reranking module, even removing some of them can still have strong performance.

4 Conclusion

In this paper, we propose a simple but effective method for question answering over knowledge graphs with triplet retrieval. Extensive experiments on two datasets show that our method can consistently outperform state-of-the-art. Ablation study further shows that our model benefits from both reranking module and the combination of sparse and dense retrievers. We believe that our work can inspire future research for question answering over knowledge graphs.

Limitations

Our work is constrained into multi-choice question answering system and limited to common sense reasoning tasks, lacking more exploration in other reasoning tasks, e.g. arithmetic reasoning (Cobbe et al., 2021; Chen et al., 2021), conversational reasoning (Chen et al., 2022) and symbolic reasoning (Wei et al., 2022). We plan to leave these directions as future work.

Ethics Statement

Our work utilizes pre-trained language model and external knowledge graph to build question answering systems. However, pre-trained language models can include biases (Shwartz and Choi, 2020) and knowledge graph, e.g. ConceptNet, has been found to contain representational harms (Mehrabi et al., 2021), which can cause these question answering systems to inherit these potential biases and harms. Therefore, additional procedures, e.g. declining inappropriate inputs and filtering harmful outputs, must be taken before real-world deployment.

References

Kurt D. Bollacker, Colin Evans, Praveen K. Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. COMET: Commonsense transformers for automatic knowledge graph construction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. *ArXiv*, abs/2210.03849.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Wenliang Dai, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*.

Jinhao Jiang, Kun Zhou, Ji-Rong Wen, and Xin Zhao. 2022. *great truths are always simple* : a rather simple knowledge encoder for enhancing the commonsense reasoning capacity of pre-trained models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1730–1741, Seattle, United States. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Shiyang Li, Jianshu Chen, and Dian Yu. 2019. Teaching pretrained models with commonsense reasoning: A preliminary kb-based approach. *ArXiv*, abs/1909.09743.

Yu Li, Baolin Peng, Yelong Shen, Yi Mao, Lars Liden, Zhou Yu, and Jianfeng Gao. 2022. Knowledge-grounded dialogue generation with a unified knowledge representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 206–218, Seattle, United States. Association for Computational Linguistics.

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, Hong Kong, China. Association for Computational Linguistics.

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*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *ArXiv*, abs/1711.05101.

Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. 2021a. Open domain question answering with a unified knowledge interface. In *Annual Meeting of the Association for Computational Linguistics*.

Xueguang Ma, Kai Sun, Ronak Pradeep, and Jimmy J. Lin. 2021b. A replication study of dense passage retriever. *ArXiv*, abs/2104.05740.

Ninareh Mehrabi, Pei Zhou, Fred Morstatter, Jay Pujara, Xiang Ren, and Aram Galstyan. 2021. Lawyers are dishonest? quantifying representational harms in commonsense knowledge resources. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5016–5033, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. 2022. SKILL: Structured knowledge infusion for large language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1581–1588, Seattle, United States. Association for Computational Linguistics.

Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Scott Yih. 2022. UniK-QA: Unified representations of structured and unstructured knowledge for open-domain question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1535–1546, Seattle, United States. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *ArXiv*, abs/1909.01066.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *Text Retrieval Conference*.

Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Timothy P. Lillicrap. 2017. A simple neural network module for relational reasoning. In *NeurIPS*.

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80.

M. Schlichtkrull, Thomas Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2017. Modeling relational data with graph convolutional networks. In *Extended Semantic Web Conference*.

Vered Shwartz and Yejin Choi. 2020. Do neural language models overcome reporting bias? In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6863–6870, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2016. Conceptnet 5.5: An open multilingual graph of general knowledge. *ArXiv*, abs/1612.03975.

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

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Kuan Wang, Yuyu Zhang, Diyi Yang, Le Song, and Tao Qin. 2022. GNN is a counter? revisiting GNN for question answering. In *International Conference on Learning Representations*.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020. 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.

Xiaoyang Wang, Pavan Kapanipathi, Ryan Musa, Mo Yu, Kartik Talamadupula, I. Abdelaziz, Maria Chang, Achille Fokoue, Bassem Makni, Nicholas Mattei, and M. Witbrock. 2018. Improving natural language inference using external knowledge in the science questions domain. *ArXiv*, abs/1809.05724.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Michihiro Yasunaga, Antoine Bosselut, Hongyu Ren, Xikun Zhang, Christopher D. Manning, Percy Liang, and Jure Leskovec. 2022. Deep bidirectional language-knowledge graph pretraining. In *Neural Information Processing Systems (NeurIPS)*.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. QA-GNN: Reasoning with language models and knowledge graphs for question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 535–546, Online. Association for Computational Linguistics.

Zhiquan Ye, Qian Chen, Wen Wang, and Zhenhua Ling. 2019. Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. *ArXiv*, abs/1908.06725.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2020. Jaket: Joint pre-training of knowledge graph and language understanding. In *AAAI Conference on Artificial Intelligence*.

Hanwen Zha, Zhiyu Chen, and Xifeng Yan. 2021. Inductive relation prediction by bert. In *AAAI Conference on Artificial Intelligence*.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022. GreaseLM: Graph REASoning enhanced language models. In *International Conference on Learning Representations*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

Appendix

A Relation mapping

We describe relation mapping with more details in Table 3, where left column shows the relation names and right column shows their corresponding templates we use to convert triplets into natural language sentences.

B More implementation and training details

We set $N = 100$ for both CommonsenseQA and OpenbookQA in passage retrieval. We set $K = 100$ and $K = 20$ for CommonsenseQA and OpenbookQA, respectively, in the reranking step. Note that for CommonsenseQA, we applied additional rules after reranking stage to filter out passages. Specifically, we filter out passages that contain the "RelatedTo" relation or do not possess any token overlaps with the answer choices. We implement the method based on huggingface `transformers` (Wolf et al., 2020), and train the models on NVIDIA A100-SXM4-40GB GPUs. For CommonsenseQA, we use the contextualized representation of first token from PLM as $\mathbf{h}$ while

relation name	relation template
Antonym	is the antonym of
AtLocation	is at location of
CapableOf	is capable of
Causes	causes
CreatedBy	is created by
IsA	is a kind of
Desires	desires
HasSubevent	has subevent
PartOf	is part of
HasContext	has context
HasProperty	has property
MadeOf	is made of
NotCapableOf	is not capable of
NotDesires	does not desire
ReceivesAction	is
RelatedTo	is related to
UsedFor	is used for
LocatedNear	is located near
CausesDesire	causes the desire of
MotivatedByGoal	is motivated by the goal of
DistinctFrom	is distinct from
HasFirstSubevent	has the first subevent
HasLastSubevent	has the last subevent
HasPrerequisite	has the prerequisite of
Entails	entails
MannerOf	a manner of
InstanceOf	an instance of
DefinedAs	is defined as
HasA	has a
SimilarTo	is similar to
Synonym	is the synonym of

Table 3: Relation name mapping for ConceptNet. We adapt this relation mapping from Yasunaga et al. (2021).

for OpenbookQA, we use the average contextualized representations of answer choice from PLM as $\mathbf{h}$. For both CommonsenseQA and OpenbookQA datasets, we use AdamW (Loshchilov and Hutter, 2017) with learning rate 10^{-5}. We set the maximum training epoch as 15 for CommonsenseQA and 10 for OpenBookQA. We set batch size as 32 and 16 for CommonsenseQA and OpenBookQA, respectively. The maximum sequence length is set to be 512 for CommonsenseQA and OpenBookQA. We run experiments with three different random seeds $\{0, 1, 2\}$ and report mean results along with standard deviations.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations section

☑ A2. Did you discuss any potential risks of your work?
Ethics Statement section

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract section

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

B ☑ Did you use or create scientific artifacts?

Section 3 Experiments and Appendix

☑ B1. Did you cite the creators of artifacts you used?
Section 3 Experiments and Appendix

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
These tools are all publicly available and we provide citations of these tools.

☒ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
We use existing artifacts consistent with their intended use.

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We use public benchmarks for experiments.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
We describe these details in section 3 Experiments.

C ☑ Did you run computational experiments?

Section 3 Experiments and Appendix

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We use RoBERTa-large that is publicly available in our paper and provide computing infrastructure. Details can be found in Section 3 Experiments and Appendix.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3 Experiments and Appendix

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3 Experiments and Appendix.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3 Experiment and Appendix

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

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.