Learning Contextualized Knowledge Structures for Commonsense Reasoning

Jun Yan
University of Southern California
yanjun@usc.edu

Mrigank Raman
Indian Institute of Technology Delhi
mt1170736@iitd.ac.in

Tianyu Zhang
Tsinghua University
zhang-ty17@mails.tsinghua.edu.cn

Ryan Rossi
Adobe Research
ryrossi@adobe.com

Handong Zhao
Adobe Research
hazhao@adobe.com

Sungchul Kim
Adobe Research
sukim@adobe.com

Nedim Lipka
Adobe Research
lipka@adobe.com

Xiang Ren
University of Southern California
xiangren@usc.edu

Abstract

Recently, neural-symbolic architectures have achieved success on commonsense reasoning through effectively encoding relational structures retrieved from external knowledge graphs (KGs). However, current methods rely on quality and contextualized knowledge structures (i.e., fact triples) retrieved at the pre-processing stage and overlook challenges caused by incompleteness of a KG (low coverage) and irrelevant retrieved facts in the reasoning context. In this paper, we present a novel neural-symbolic approach, named Hybrid Graph Network (HGN), which jointly generates feature representations for new triples (as a complement to existing edges in the KG), determines the relevance of the triples to the reasoning context, and learns graph model parameters for reasoning. Our method learns a compact graph structure through filtering edges that are unhelpful to reasoning. We show marked improvements on three commonsense reasoning benchmarks.

1 Introduction

Commonsense knowledge plays a vital role in human communication, but the fact that relevant commonsense knowledge is rarely mentioned explicitly makes it hard for neural models to achieve human-level language understanding. Fig. 1 shows an example in a commonsense reasoning benchmark named CommonsenseQA [31], where external relational knowledge about concepts is required to infer the answer. Existing commonsense reasoning frameworks can be classified into information retrieval-augmented (IR-augmented) methods [2, 24] and knowledge-graph-augmented (KG-augmented) methods [34, 12]. We focus on KG-augmented methods as KGs provide *structured* relational knowledge between concepts, making them a good fit for tasks that require reasoning.

The core challenges of incorporating KG knowledge into context comprehension are how to obtain related evidence–*i.e.*, a *contextualized* knowledge graph of inter-related fact triples; and how to reason over the graph. Most existing works [16, 34, 19] simply extract triples from KGs based on the concepts mentioned in the context, which is far from ideal as KGs are likely incomplete [22] and helpful facts could be missing. What's worse, since KGs are context-agnostic, the extracted facts don't necessarily relate to the sentence's central topic, and could thus be distracting.

To address these issues, we propose Hybrid Graph Network (HGN), a novel framework for KG-augmented commonsense reasoning. It leverages both extracted facts (with high precision) and generated facts (with high recall) by building a contextualized knowledge graph with hybrid (extracted

4th Knowledge Representation and Reasoning Meets Machine Learning Workshop (KR2ML 2020), at NeurIPS.

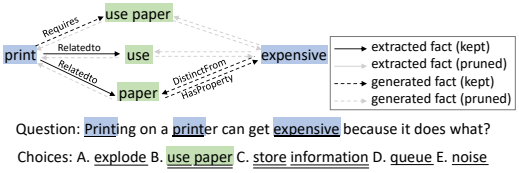


Figure 1: **Commonsense question answering augmented with external knowledge.** Underlined words and phrases are recognized concepts.

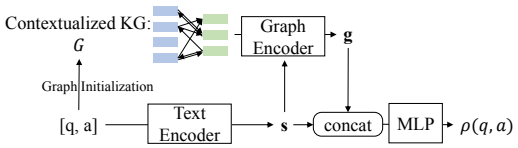


Figure 2: **Architecture of a typical neural-symbolic model for commonsense reasoning.**

+ generated) edge features. It then dynamically prunes *unreliable* and *unrelated* edges, leading to a superior graph structure for reasoning. Fig. 1 presents an example of the learned graph structure with hybrid edge features, where critical evidence triples are generated, and edges between concepts that are not closely related are pruned. We conduct extensive experiments on three commonsense reasoning benchmarks and show consistent improvement over previous KG-augmented methods.

2 Methodology

We focus on the task of commonsense question answering (QA), while the proposed framework can be easily applied to other tasks that require commonsense reasoning. Given a question q, the model is asked to select the correct answer from a set of candidate answers $\{a_i\}$ with the help of symbolic knowledge from an external knowledge graph $\mathcal{G} = \{\mathcal{E}, \mathcal{R}, \mathcal{F}\}$, where $\mathcal{E}, \mathcal{R}, \mathcal{F}$ denote the set of entities, relations, and facts (triples), respectively. A fact takes the form of a triple $(h, r, t) \in \mathcal{F}$, where $h, t \in \mathcal{E}$ are the head and tail entities, and $r \in \mathcal{R}$ is their relation.

We approach multiple-choice QA by measuring the *plausibility* $\rho(q, a)$ between a question q and a candidate answer a, where the candidate answer with the highest plausibility score will be chosen. Fig. 2 illustrates the workflow of a neural-symbolic architecture for QA. The final score is predicted based on the unstructured textual evidence and structured graph evidence. We adopt a pretrained language model as the text encoder to get the statement vector: $\mathbf{s} = f_{text}([q, a])$, where $[\cdot, \cdot]$ denotes sentence concatenation. For graph evidence, we build a directed *contextualized knowledge graph* $G = (V, E)$ with adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$, which stores structured knowledge related to the question and answer context for reasoning. We tackle the challenge of missing facts by densifying an extracted graph with generated facts. We resolve the issue with noisy facts by jointly pruning the graph structure and learning network parameters (see an overview of our approach in Fig. 3).

We denote the label for (q, a) as y, where $y = 1$ means a is the correct answer to q and $y = 0$ means a is a wrong answer. Then the overall learning objective on the training set D_{train} is defined as:

$$\mathcal{L} = \sum\nolimits_{(q,a,y) \in D_{train}} \left[L_{task}(q, a, y) + \beta \cdot L_{prune}(\mathbf{A}^K) \right], \tag{1}$$

where β is a hyperparameter, $\mathbf{A}^K$ represents the final graph structure after K layers' refinement. $\mathcal{L}$ can be decomposed into L_{task} for the classification task and L_{prune} for graph structure learning.

Graph Initialization. To effectively acquire knowledge from $\mathcal{G}$, we perform string matching to ground concepts in (q, a) to the entity set of $\mathcal{G}$. $V = V^Q \cup V^A$ where V^Q and V^A are the sets of recognized concepts in q and a. Node feature vectors $\{\mathbf{x}_i\}$ are initialized using TransE [4] embeddings. We consider all directed edges between question concepts and answer concepts: $E = (V^A \times V^Q) \cup (V^Q \times V^A)$, which serve as discriminative features in evaluating the plausibility. Note that for a large portion of $(v_i, v_j) \in E$, there may not be any fact from $\mathcal{G}$ that describes their relation. We therefore turn to use an edge feature generator $f_{gen}(\cdot, \cdot)$ to capture their relational feature. Implementation details can be found in §A. For any $(v_i, v_j) \in E$, the edge feature $\mathbf{x}_{(i,j)}$ is calculated in a hybrid way. When there exists $r \in \mathcal{R}$ such that $(v_i, r, v_j) \in \mathcal{F}$, $\mathbf{x}_{(i,j)} = \mathbf{r}$ where $\mathbf{r}$ is the learnable embedding of relation r. Otherwise, $\mathbf{x}_{(i,j)} = f_{gen}(v_i, v_j)$.

Reasoning with Weighted Graph. Although we use an edge feature generator to densify the connections between question and answer concepts, we are still facing the challenges of unreliable edges (edges with wrong or low-quality attributes) and unrelated edges (edges irrelevant to answering the question) in the contextualized graph G. To account for the quality difference among different edge features, we employ a rescaling operation on each edge to control its information flow. The rescaling factor can also be interpreted as the edge weight, which indicates the helpfulness of a certain edge in reasoning. We build our model based on the formulation of Graph Networks (GNs) [3] by instantiating the node-to-edge ($v \to e$) and edge-to-node ($e \to v$) message passing functions for each layer. The propagation rule at layer l is defined as follows.

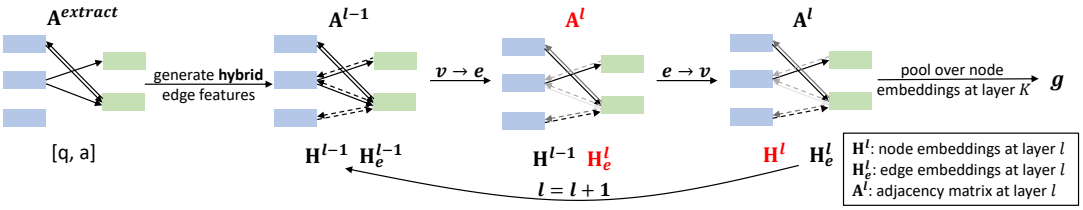


Figure 3: **Overview of our HGN model. We jointly learn the graph structure and network parameters.** Darkness of edges indicate their weights. Red variables are updated in the previous step.

$$v \to e : \mathbf{h}^l_{(i,j)} = f^l_{v\to e}\left(\left[\mathbf{h}_i^{l-1}; \mathbf{h}_j^{l-1}; \mathbf{h}^{l-1}_{(i,j)}; \mathbf{s}\right]\right); \quad w^l_{(i,j)} = f^l_w\left(\left[\mathbf{h}^{l-1}_{(i,j)}; \mathbf{s}\right]\right); \quad \mathbf{A}^l_{(i,j)} = \frac{e^{w^l_{(i,j)}}}{\sum_{(s,t)\in E} e^{w^l_{(s,t)}}},$$
$$e \to v : \mathbf{u}^l_{(i,j)} = f^l_u\left(\left[\mathbf{h}_i^{l-1}; \mathbf{h}^l_{(i,j)}\right]\right); \quad \mathbf{h}^l_j = f^l_{e\to v}\left(\sum\nolimits_{i\in\mathcal{N}_j} \mathbf{A}^l_{(i,j)}\mathbf{u}^l_{(i,j)}\right). \tag{2}$$

Here N_j is the set of v_j's neighboring nodes, $f^l_{v\to e}$, f^l_u, f^l_w and $f^l_{e\to v}$ are implemented as different multilayer perceptrons (MLPs), $\mathbf{h}^0_{(i,j)} = \mathbf{x}_{(i,j)}$, $\mathbf{h}^0_i = \mathbf{x}_i$, $\mathbf{A}^0 = \mathbf{A}$. Basically, for each edge $(v_i, v_j) \in E$, we learn a normalized weight $\mathbf{A}^l_{(i,j)}$. Therefore, our model can learn to "softly" prune an edge by assigning to it a weight that is close to 0. We use attentive pooling to summarize the final layer's hidden representations of all nodes as the graph vector: $\alpha_i = \mathbf{s}\mathbf{W}_{att}\mathbf{h}^K_i$, $\mathbf{g} = \sum_{i:v_i\in V} \frac{e^{\alpha_i}}{\sum_{j:v_j\in V} e^{\alpha_j}}\mathbf{h}^K_i$, where $\mathbf{W}_{att}$ is a learnable matrix for calculating the attention score α_i for each node v_i. We concatenate $\mathbf{s}$ and $\mathbf{g}$ and use an MLP follwed by a softmax layer to calculate the probability for each candidate answer: $\rho(q,a) = f_{MLP}([\mathbf{s};\mathbf{g}])$, $\{\hat{\rho}(q,a_i)\} = \mathsf{softmax}\{\rho(q,a_i)\}$.

Pruning with Entropy Regularization. To encourage the model to take decisive pruning steps on the graph structure, we add a regularization term to the loss function to penalize non-discriminative edge weights. In an extreme case, a blind model will assign the same weight to all edges, where usually noisy edges are mixed with helpful edges. To avoid that, we minimize the entropy of the edge weight distribution as an auxiliary training objective. The motivation is that information entropy can be used to measure the informativeness of the edge weight predictions. A lower entropy, caused by a skewed distribution, means the model is incorporating more priors (*e.g.* the plausibility of its corresponding fact and the reasoning context) into edge weight prediction. Formally, the entropy regularization term of G is calculated as: $L_{prune}(\mathbf{A}^K) = -\sum_{(i,j):(v_i,v_j)\in E} \mathbf{A}^K_{(i,j)} \log \mathbf{A}^K_{(i,j)}$.

Model Training. We adopt the cross-entropy loss for the main classification task and add the penalty term so that the graph structure can be jointly learned with graph reasoning. Then the overall learning objective on the training set D_{train} (Eq. 1) is derived as:

$$\mathcal{L} = \sum\nolimits_{(q,a,y)\in D_{train}} \left[-y\log\hat{\rho}(q,a) - \beta\sum\nolimits_{(i,j):(v_i,v_j)\in E} \mathbf{A}^K_{(i,j)} \log \mathbf{A}^K_{(i,j)}\right]. \tag{3}$$

We train the whole model end-to-end by minimizing $\mathcal{L}$ using RAdam [17] optimizer.

3 Experiments

Experimental Setup. We evaluate our proposed framework on three multiple-choice commonsense QA datasets: **CommonsenseQA** [31], **OpenbookQA** [21] and **CODAH** [5] (details in §B). We use ConceptNet [30] as knowledge graph $\mathcal{G}$. For f_{text}, we experiment with BERT-base, BERT-large [6] and RoBERTa (-large) [18] to validate our model's effectiveness over different text encoders. For OpenbookQA, we also build our model on top of an IR-augmented method named "AristoRoBERTa" to study if strong IR-augmented methods could still benefit from KG knowledge and our reasoning method. We compare our model with KG-augmented methods including **RN** [27], **RN + Link Prediction**, **RGCN** [28], **GN** [3], **GconAttn** [34], **KagNet** [16], **MHGRN** [7], **PathGenerator** [32]. Details can be found in §C.

Results. Tables 1, 2[1] and 3 show performance comparisons between our models and baseline models on CommonsenseQA, OpenbookQA, and CODAH respectively. Our HGN shows consistent improvement over baseline models on all datasets except that it achieves the second-best performance on OpenbookQA with AristoRoBERTa as the text encoder. We also submitted our best model to OpenbookQA's leaderboard.[2] Our model ranks the first among all models using AristoRoBERTa

[1] Some of the baseline results on CommonsenseQA and OpenbookQA are reported by Feng et al. [7] and Wang et al. [32]. Mean and standard deviation of four runs are presented for all models except KagNet.

[2] We tried more than 4 seeds for the leaderboard submission, which is different from the setting of Table 2.

Methods	BERT-Base		BERT-Large		RoBERTa	
	60% Train	100% Train	60% Train	100% Train	60% Train	100% Train
LM Finetuning	52.06(±0.72)	53.47 (±0.87)	52.30 (±0.16)	55.39 (±0.40)	65.56 (±0.76)	68.69 (±0.56)
RN [27]	54.43 (±0.10)	56.20 (±0.45)	54.23 (±0.28)	58.46 (±0.71)	66.16 (±0.28)	70.08 (±0.21)
RN + Link Prediction	-	-	53.96 (±0.56)	56.02 (±0.55)	66.29(±0.29)	69.33 (±0.98)
RGCN [29]	52.20 (±0.31)	54.50 (±0.56)	54.71 (±0.37)	57.13 (±0.36)	68.33 (±0.85)	68.41 (±0.66)
GN [3]	53.67 (±0.45)	55.65 (±0.51)	54.78 (±0.61)	57.81 (±0.67)	68.78(±0.67)	71.12 (±0.45)
GconAttn [33]	51.36 (±0.98)	54.41 (±0.50)	54.96 (±0.69)	56.94 (±0.77)	68.09(±0.63)	69.88 (±0.47)
KagNet [16]	-	56.19	-	57.16	-	-
MHGRN [7]	54.12 (±0.49)	56.23 (±0.82)	56.76 (±0.21)	59.85 (±0.56)	68.84(±1.06)	71.11 (±0.81)
PathGenerator [32]	-	-	55.47 (±0.92)	57.21 (±0.45)	68.65(±0.02)	71.55 (±0.99)
HGN	**55.39** (±0.34)	**57.82** (±0.23)	**57.23**(±0.56)	**60.43** (±0.54)	**70.34** (±0.79)	**72.88** (±0.83)

Table 1: **Accuracy on CommonsenseQA inhouse test set.** We use the inhouse split as Lin et al. [16].

Methods	RoBERTa	AristoRoBERTa
LM Finetuning	64.80 (±2.37)	77.40 (±1.64)
RN [27]	63.65 (±2.31)	75.35 (±1.39)
RN + Link Prediction	66.30 (±0.48)	77.25 (±1.11)
RGCN [29]	62.45 (±1.57)	74.60 (±2.53)
GN [3]	66.20 (±2.14)	77.25 (±0.91)
GconAttn [33]	64.75 (±1.48)	71.80 (±1.21)
MHGRN [7]	66.85 (±1.19)	77.75 (±0.38)
PathGenerator [32]	68.40 (±0.31)	**80.05** (±0.68)
HGN	**69.00** (±0.95)	79.00 (±1.43)

Table 2: **Test accuracy on OpenbookQA.**

Methods	BERT-large	RoBERTa
LM Finetuning	65.89	83.20
RN [27]	66.24	82.85
RGCN [29]	65.27	82.24
GN [3]	65.89	82.24
MHGRN [7]	66.17	83.21
HGN	**66.71**	**83.75**

Table 3: **Test accuracy on CODAH.** We use 5-fold cross validation with the same split as Yang et al. [35].

as the text encoder (see §D), demonstrating the effectiveness of our proposed approach. We notice that most baseline models fail to achieve improvement over AristoRoBERTa. That may be because AristoRoBERTa has access to knowledge collected through IR, making it difficult to further benefit from KG knowledge if a weak reasoning approach is adopted. The improvement on CODAH is less significant compared to the other two datasets. As Chen et al. [5] suggest, questions in CODAH mainly target commonsense reasoning about quantitative, negation and object reference. In this case, relational knowledge provided by ConceptNet may only offer limited help. We further perform ablation studies and a user study on the refined graph structures, which are presented in §E and §F.

4 Related Work

KG-Augmented Commonsense Question Answering. Literature in this domain mainly studies how to encode the contextualized graph extracted from KG. For example, Lin et al. [16] propose a model comprised of GCN and LSTM to account for both the global graph structure and local paths connecting question concepts and answer concepts. Ma et al. [20] use BERT to generate the embedding for the pseudo-sentence representing each edge and pool over edge features to get the graph encoding. Crucial difference is that they assume a static graph and there's no operation on enriching or denoising the graph structure. While Wang et al. [32] build the contextualized graph with a path generator, they reason over a static graph and neglect the noise introduced during generation.

Graph Structure Learning. Works that jointly learn the graph structure with the downstream task can be classified into two categories. One line of works directly learn an unweighted graph with desired edges for reasoning. Kipf et al. [14] and Franceschi et al. [9] sample the graph structure from a predicted probabilistic distribution with differentiable approximations. Norcliffe-Brown et al. [23] calculate the relatedness between any pair of nodes and only keep the top-k strongest connections for each node to construct the edge set. The other line of works consider a weighted graph with all possible edges and softly filter out the noisy ones by downweighting them. An adjacency matrix with continuous values is incorporated into message passing. Jiang et al. [11] and Yu et al. [36] use heuristics to regularize the learned adjacency matrix. Hu et al. [10] consider the question embedding for predicting edge weights. Our HGN falls into the second category and therefore avoids information loss caused by hard pruning and approximation. Our uniqueness is that we construct the graph with hybrid features based on extracted and generated facts and we let node features, edge features, edge weights, and the global signal (statement vector) collectively determine the evolution of the graph structure. These empower our model with greater capacity and flexibility for KG-augmented QA.

5 Conclusion

In this paper, we propose a neural-symbolic framework for commonsense reasoning named HGN. To address the issues with missing facts in the external knowledge graph and noisy facts in the contextualized knowledge graph, our proposed HGN jointly generates features for new edges, refines the graph structure, and learns the parameters for graph networks. Experimental results on three commonsense reasoning benchmarks demonstrate the effectiveness of our model.

References

[1] P. Banerjee and C. Baral. Knowledge fusion and semantic knowledge ranking for open domain question answering. *arXiv preprint arXiv:2004.03101*, 2020.

[2] P. Banerjee, K. K. Pal, A. Mitra, and C. Baral. Careful selection of knowledge to solve open book question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6120–6129, 2019.

[3] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[4] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795, 2013.

[5] M. Chen, M. D'Arcy, A. Liu, J. Fernandez, and D. Downey. Codah: An adversarially authored question-answer dataset for common sense. *arXiv preprint arXiv:1904.04365*, 2019.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Y. Feng, X. Chen, B. Y. Lin, P. Wang, J. Yan, and X. Ren. Scalable multi-hop relational reasoning for knowledge-aware question answering. *arXiv preprint arXiv:2005.00646*, 2020.

[8] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76 (5):378, 1971.

[9] L. Franceschi, M. Niepert, M. Pontil, and X. He. Learning discrete structures for graph neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97. PMLR, 2019.

[10] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko. Language-conditioned graph networks for relational reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10294–10303, 2019.

[11] B. Jiang, Z. Zhang, D. Lin, J. Tang, and B. Luo. Semi-supervised learning with graph learning-convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11313–11320, 2019.

[12] P. Kapanipathi, V. Thost, S. S. Patel, S. Whitehead, I. Abdelaziz, A. Balakrishnan, M. Chang, K. P. Fadnis, R. C. Gunasekara, B. Makni, et al. Infusing knowledge into the textual entailment task using graph convolutional networks. In *AAAI*, pages 8074–8081, 2020.

[13] D. Khashabi, T. Khot, A. Sabharwal, O. Tafjord, P. Clark, and H. Hajishirzi. Unifiedqa: Crossing format boundaries with a single qa system. *arXiv preprint arXiv:2005.00700*, 2020.

[14] T. Kipf, E. Fetaya, K.-C. Wang, M. Welling, and R. Zemel. Neural relational inference for interacting systems. *arXiv preprint arXiv:1802.04687*, 2018.

[15] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[16] B. Y. Lin, X. Chen, J. Chen, and X. Ren. 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 2822–2832, 2019.

[17] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the variance of the adaptive learning rate and beyond. In *Proceedings of the Eighth International Conference on Learning Representations (ICLR 2020)*, April 2020.

[18] 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.

[19] 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*, pages 8449–8456, 2020.

[20] K. Ma, J. Francis, Q. Lu, E. Nyberg, and A. Oltramari. Towards generalizable neuro-symbolic systems for commonsense question answering. *EMNLP 2019*, page 22, 2019.

[21] 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 *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, 2018.

[22] B. Min, R. Grishman, L. Wan, C. Wang, and D. Gondek. Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 777–782, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1095.

[23] W. Norcliffe-Brown, S. Vafeias, and S. Parisot. Learning conditioned graph structures for interpretable visual question answering. In *Advances in neural information processing systems*, pages 8334–8343, 2018.

[24] X. Pan, K. Sun, D. Y. J. Chen, H. Ji, C. Cardie, and D. Yu. Improving question answering with external knowledge. In *EMNLP 2019 MRQA Workshop*, page 27, 2019.

[25] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. Language models are unsupervised multitask learners. 2019.

[26] 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. *arXiv preprint arXiv:1910.10683*, 2019.

[27] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.

[28] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer, 2018.

[29] M. S. Schlichtkrull, T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, and M. Welling. Modeling relational data with graph convolutional networks. In A. Gangemi, R. Navigli, M. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, editors, *The Semantic Web - 15th International Conference, ESWC 2018, Heraklion, Crete, Greece, June 3-7, 2018, Proceedings*, volume 10843 of *Lecture Notes in Computer Science*, pages 593–607. Springer, 2018. doi: 10.1007/978-3-319-93417-4_38. URL https://doi.org/10.1007/978-3-319-93417-4_38.

[30] R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: an open multilingual graph of general knowledge. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 4444–4451, 2017.

[31] A. Talmor, J. Herzig, N. Lourie, and J. Berant. 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, 2019.

[32] P. Wang, N. Peng, P. Szekely, and X. Ren. Connecting the dots: A knowledgeable path generator for commonsense question answering. *arXiv preprint arXiv:2005.00691*, 2020.

[33] X. Wang, P. Kapanipathi, R. Musa, M. Yu, K. Talamadupula, I. Abdelaziz, M. Chang, A. Fokoue, B. Makni, N. Mattei, and M. Witbrock. Improving natural language inference using external knowledge in the science questions domain. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7208–7215. AAAI Press, 2019. doi: 10.1609/aaai.v33i01.33017208. URL https://doi.org/10.1609/aaai.v33i01.33017208.

[34] 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7208–7215, 2019.

[35] Y. Yang, C. Malaviya, J. Fernandez, S. Swayamdipta, R. L. Bras, J.-P. Wang, C. Bhagavatula, Y. Choi, and D. Downey. G-daug: Generative data augmentation for commonsense reasoning. *arXiv preprint arXiv:2004.11546*, 2020.

[36] D. Yu, R. Zhang, Z. Jiang, Y. Wu, and Y. Yang. Graph-revised convolutional network. *arXiv preprint arXiv:1911.07123*, 2019.

A Edge Feature Generator

As an implementation of f_{gen}, we adopt GPT-2 [25], which is pretrained on large corpora and achieves great success on a wide range of tasks involving sentence generation, as a generator to generalize the facts from the knowledge graph. We first convert each fact $(h, r, t) \in \mathcal{F}$ into a word sequence with a "prompt-generation" format: $\left[\tilde{h}; \$; \tilde{t}; \$; \tilde{h}; \tilde{r}; \tilde{t}\right]$, where $\tilde{h}, \tilde{r}, \tilde{t}$ are the word sequence of h, r, t respectively, $\$$ denotes the delimiter token used by GPT-2, and $[\cdot;\cdot]$ denotes word sequence concatenation. We denote the synthetic sentence as $s_{(h,r,t)} = \left[x_1^{(h,r,t)}, \ldots, x_{n_{(h,r,t)}}^{(h,r,t)}\right]$ and finetune GPT-2 on all synthetic sentences created from $\mathcal{F}$ with the language modeling objective:

$$L_{gen}(\mathcal{F}) = \sum_{(h,r,t)\in\mathcal{F}} \sum_{i=1}^{n_{(h,r,t)}} \log P\left(x_i^{(h,r,t)} \mid x_1^{(h,r,t)}, \ldots, x_{i-1}^{(h,r,t)}\right). \tag{4}$$

After that, given any two concepts (v_i, v_j), we build a prompt as $[\tilde{v_i}; \$; \tilde{v_j}; \$]$ and let the model to generate the following word sequence. We denote the whole sentence (both prompt and generation) as $s_{(v_i,v_j)}$, and the hidden states of each word during generation as $\mathbf{h}_1, \ldots, \mathbf{h}_T$ where T is the sentence length. To get the relational feature, we apply a learnable linear transformation to the averaged hidden states of all words in the sentence: $f_{gen}(v_i, v_j) = \mathbf{W}\left[\frac{1}{T}\sum_{i=1}^{T}\mathbf{h}_i\right] + \mathbf{b}$ and ensure that $f_{gen}(v_i, v_j)$ has the same dimension as $\mathbf{r}$.

B Datasets

CommonsenseQA [31] is a multiple-choice QA dataset targeting commonsense. It's constructed based on the knowledge in ConceptNet. Since the test set of the official split (9741/1221/1140 for OFtrain/OFdev/OFtest) is not publicly available, we compare our models with baseline models on the inhouse split (8500/1221/1241 for IHtrain/IHdev/IHtest) used by previous works [16, 7, 32].

OpenbookQA [21] is a multiple-choice QA dataset modeled after openbook exams. Besides 5957 elementary-level science questions (4957/500/500 for train/dev/test), it also provides an open book with 1326 core science facts. Solving the dataset requires combining facts from open book with commonsense knowledge.

CODAH [5] contains 2801 sentence completion questions testing commonsense reasoning skills. We perform 5-fold cross validation using the same split as Yang et al. [35].

C Compared Methods

RN [27] builds the graph with the same node set as our method but extracted edges only. The graph vector is calculated as $g = \text{Pool}(\{\text{MLP}([\mathbf{x}_i; \mathbf{x}_{(i,j)}; \mathbf{x}_j]) \mid (v_i, r, v_j) \in \mathcal{F}\})$. **RN + Link Prediction** differs from RN by only considering the generated relation between every question and answer concepts. The relation is predicted using knowledge graph completion method based on TransE [4] embeddings. **RGCN** [28] extends Graph Convolutional Networks (GCNs) [15] with relation-specific transition matrices during message passing. It operates on the same graph as RN. The graph vector is calculated as $\mathbf{g} = \text{Pool}(\{\mathbf{h}_i^K \mid v_i \in V\})$. **GN** [3] presents a general formulation of GNNs. We instantiate it with the layerwise propagation rule defined in Eq. 2. It differs from our HGN in that: (1) it only considers extracted edges; (2) all edge weights are fixed to 1. **GconAttn** [34] softly aligns the nodes in question and answer and do pooling over all matching nodes to get $\mathbf{g}$. **KagNet** [16] uses an LSTM to encode relational paths between question and answer concepts and pool over the path embeddings for graph encoding. **MHGRN** [7] generalizes GNNs with multi-hop message passing. **PathGenerator**[3][4] [32] learns a path generator and pool over embeddings for all generated paths connecting question and answer concepts.

[3] PathGenerator is a contemporaneous work that learns a edge feature generator based on multi-hop paths, which has greater expressive power than our (1-hop) edge feature generator. Our reasoning framework is compatible with any implementation for f_{gen}, and we will build on PathGenerator in our future experiments.

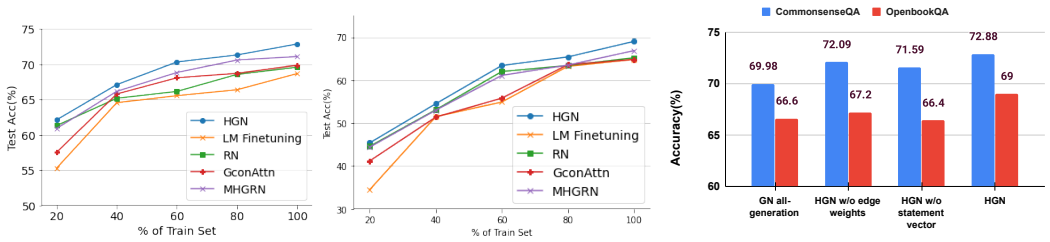
[4] We choose PG-Global as the representative variant for PathGenerator as it performs better than PG-Local. PG-Full is an ensemble model of PG-Global and RN, so we don't consider it in our comparisons.

D Leaderboard of OpenbookQA

Methods	Text Encoder	Test Acc
UnifiedQA [13]	T5	87.2
T5 + KB	T5	85.4
T5 [26]	T5	83.2
PathGenerator [32]	AristoAlbert	81.8
HGN (ours)	**AristoRoBERTa**	**81.4**
AristoRoBERTa + KB	AristoRoBERTa	81.0
MHGRN [7]	AristoRoBERTa	80.6
PathGenerator [32]	AristoRoBERTa	80.2
KF + SIR [1]	RoBERTa	80.2
AristoRoBERTa	AristoRoBERTa	80.2

Table 4: **Leaderboard of OpenbookQA.** Our HGN ranks first among all submissions using AristoRoBERTa as the text encoder.

E Ablation Studies



(a) CommonsenseQA. (b) OpenbookQA. (c) Ablations on model variants.

Figure 4: **Ablation studies.** (a)(b) Performance of HGN and baseline models with different amounts of training data; (c) Performance of different model variants.

Training with less labeled data. Fig. 4 (a)(b) show the results of our model and baseline models when trained with 20%/40%/60%/80%/100% of the training data on CommonsenseQA and OpenbookQA. Our model gets better test accuracy under all settings. The improvement over the knowledge-agnostic baseline (LM-finetuning) is more significant with less training data, which suggests that incorporating external knowledge is more helpful in the low-resource setting.

Study on more model variants. To better understand the model design, we experiment with two more variants on CommonsenseQA and OpenbookQA. **GN all-generation** doesn't consider extracted facts and instead generate edge features between all question and answer concepts. It then runs GN over the graph. **HGN w/o statement vector** doesn't consider $\mathbf{s}$ in Eq. 2, which isolates the graph encoder from the text encoder. Fig. 4 (c) shows the results of the ablation study. Comparing "GN all-generation" with "HGN w/o edge weights", we can conclude that extracted facts play an important role in HGN and can't be replaced by generated features. The high precision of extracted facts is still desirable even if we have a model to generate relational edges. Comparing "HGN w/o statement vector" with "HGN", we find that accessing context information is also important for graph reasoning, which means information propagation and edge weight prediction should be conducted in a context-aware manner.

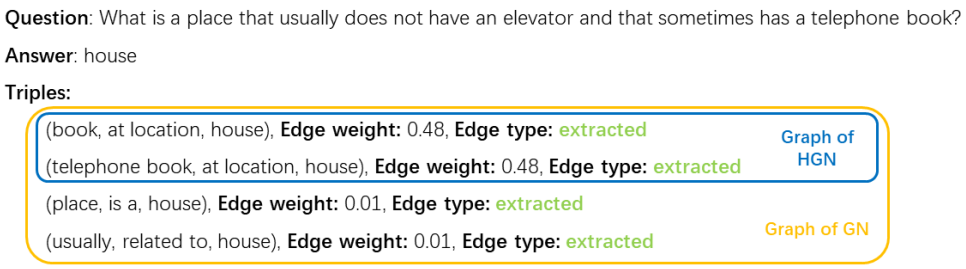
F User Study

To assess our model's ability to refine the graph structure, we compare the graph structure before and after being processed by HGN. Specifically, we sample 30 questions with its correct answer from the development set of CommonsenseQA and ask 5 human annotators to evaluate the graph output by GN (with adjacency matrix $A^{extract}$ and extracted facts only) and our HGN (with adjacency matrix A^K). We manually binarizing A^K by removing edges with weight less than 0.01.

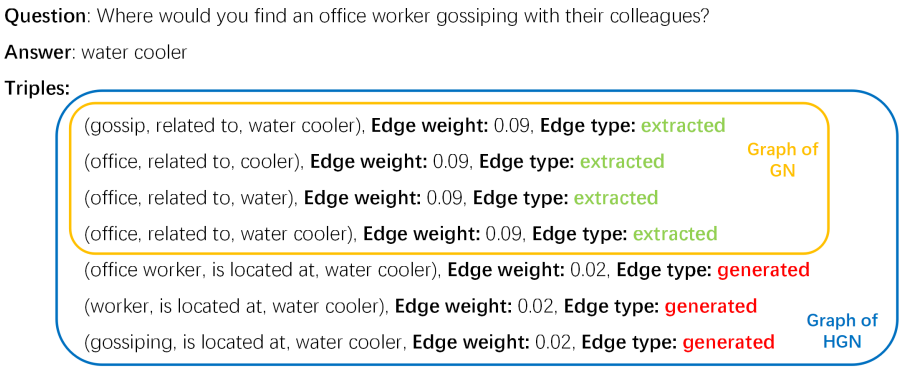
Given a graph, for each edge (fact), annotators are asked to rate its **validness** and **helpfulness**. The validness score is rated as a binary value in a context-agnostic way: 0 (the fact doesn't make sense), 1 (the fact is generally true). The helpfulness score measures if the fact is helpful for solving the question and is rated on a 0 to 2 scale: 0 (the fact is unrelated to the question and answer), 1 (the fact is related but doesn't directly lead to the answer), 2 (the fact directly leads to the answer). The mean ratings for 30 pairs of (GN, HGN) graphs by 5 annotators are reported in Table 5. We also include another metric named "prune rate" calculated as: $1 - \frac{\text{\# edges in binarized } A^K}{\text{\# edges in } A^0}$, which measures the portion of edges that are assigned very low weights (softly pruned) during training and is only applicable to HGN. The Fleiss' Kappa [8] is 0.51 (moderate agreement) for validness and 0.36 (fair agreement) for helpfulness. The graph refined by HGN has both more edges and more valid edges compared to the extracted one. The refined graph also achieves a higher helpfulness score. These all indicate that our HGN learns a superior graph structure with more helpful edges and less noisy edges, which is the reason for performance improvement over previous works that rely on extracted and static graphs. Two cases comparing the extracted graph ("Graph of GN") with our refined graph ("Graph of HGN") are showed in Fig. 5.

Contextualized Graph	**GN** ($\mathbf{A}^{extract}$)	**HGN** ($\mathbf{A}^{K}$)
Number of Edges	3.65 ($\pm$2.73)	4.38 ($\pm$3.24)
Number of Valid Edges	2.67 ($\pm$1.95)	3.15 ($\pm$1.98)
Percentage of Valid Edges	71.64%	78.51%
Average Helpfulness Score of Edges	0.90 ($\pm$0.50)	1.16 ($\pm$0.51)
Prune Rate	-	77.13%

Table 5: **User study on learned graph structures.** 30 pairs of contextualized graphs output by GN and HGN are evaluated by 5 annotators.



(a) Case I: HGN prunes unhelpful edges in the extracted graph.



(b) Case II: HGN generates helpful new edges.

Figure 5: **Two cases showing the graphs output by GN and HGN.**