QA-GNN: Reasoning with Language Models and Knowledge Graphs for Question Answering

Michihiro Yasunaga **Hongyu Ren** **Antoine Bosselut**
Percy Liang **Jure Leskovec**
Stanford University
{myasu,hyren,antoineb,pliang,jure}@cs.stanford.edu

Abstract

The problem of answering questions using knowledge from pre-trained language models (LMs) and knowledge graphs (KGs) presents two challenges: given a QA context (question and answer choice), methods need to (i) identify relevant knowledge from large KGs, and (ii) perform joint reasoning over the QA context and KG. Here we propose a new model, *QA-GNN*, which addresses the above challenges through two key innovations: (i) relevance scoring, where we use LMs to estimate the importance of KG nodes relative to the given QA context, and (ii) joint reasoning, where we connect the QA context and KG to form a joint graph, and mutually update their representations through graph-based message passing. We evaluate QA-GNN on the CommonsenseQA and OpenBookQA datasets, and show its improvement over existing LM and LM+KG models, as well as its capability to perform interpretable and structured reasoning, *e.g.*, correctly handling negation in questions.

1 Introduction

Question answering systems must be able to access relevant knowledge and reason over it. Typically, knowledge can be implicitly encoded in large language models (LMs) pre-trained on unstructured text (Petroni et al., 2019; Bosselut et al., 2019), or explicitly represented in structured knowledge graphs (KGs), such as Freebase (Bollacker et al., 2008) and ConceptNet (Speer et al., 2017), where entities are represented as nodes and relations between them as edges. Recently, pre-trained LMs have demonstrated remarkable success in many question answering tasks (Liu et al., 2019; Raffel et al., 2020). However, while LMs have a broad coverage of knowledge, they do not empirically perform well on structured reasoning (*e.g.*, handling negation) (Kassner and Schütze, 2020). On the other hand, KGs are more suited for structured reasoning (Ren et al., 2020; Ren and Leskovec, 2020) and enable explainable predictions *e.g.*, by providing reasoning paths (Lin et al., 2019), but may lack coverage and be noisy (Bordes et al., 2013; Guu et al., 2015). How to reason effectively with both sources of knowledge remains an important open problem.

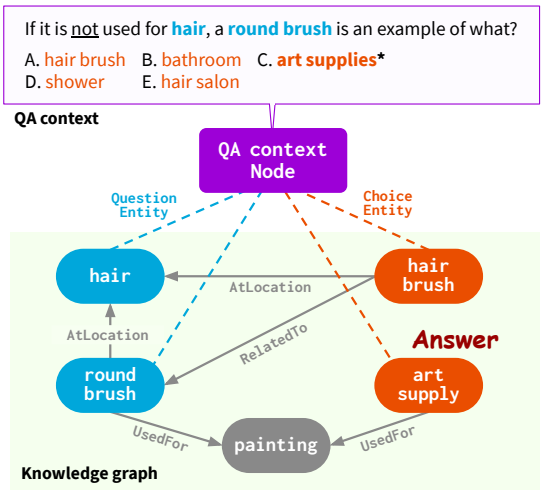

Figure 1: Given the QA context (question and answer choice; purple box), we aim to derive the answer by performing joint reasoning over the language and the knowledge graph (green box).

Combining LMs and KGs for reasoning (henceforth, LM+KG) presents two challenges: given a QA context (*e.g.*, question and answer choices; Figure 1 purple box), methods need to (i) identify informative knowledge from a large KG (green box); and (ii) capture the nuance of the QA context and the structure of the KGs to perform joint reasoning over these two sources of information. Previous works (Bao et al., 2016; Sun et al., 2018; Lin et al., 2019) retrieve a subgraph from the KG by taking *topic entities* (KG entities mentioned in the given QA context) and their few-hop neighbors. However, this introduces many entity nodes that are semantically irrelevant to the QA context, especially when the number of topic entities or hops increases. Additionally, existing LM+KG methods for reasoning (Lin et al., 2019; Wang et al., 2019a; Feng et al., 2020; Lv et al., 2020) treat the QA context and KG as two separate modalities. They individually apply LMs to the QA context and graph

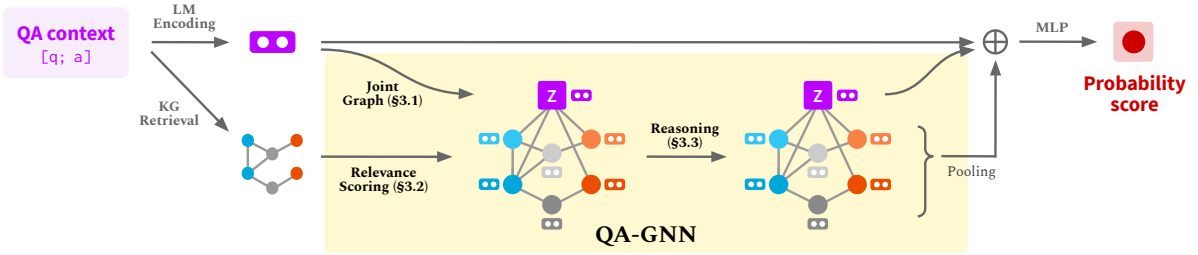

Figure 2: Overview of our approach. Given a QA context (z), we connect it with the retrieved KG to form a joint graph (*working graph*; §3.1), compute the relevance of each KG node conditioned on z (§3.2; node shading indicates the relevance score), and perform reasoning on the working graph (§3.3).

neural networks (GNNs) to the KG, and do not mutually update or unify their representations. This separation might limit their capability to perform structured reasoning, *e.g.*, handling negation.

Here we propose *QA-GNN*, an end-to-end LM+KG model for question answering that addresses the above two challenges. We first encode the QA context using an LM, and retrieve a KG subgraph following prior works (Feng et al., 2020). Our QA-GNN has two key insights: (i) **Relevance scoring**: Since the KG subgraph consists of all few-hop neighbors of the topic entities, some entity nodes are more relevant than others with respect to the given QA context. We hence propose KG node relevance scoring: we score each entity on the KG subgraph by concatenating the entity with the QA context and calculating the likelihood using a pre-trained LM. This presents a general framework to weight information on the KG; (ii) **Joint reasoning**: We design a joint graph representation of the QA context and KG, where we explicitly view the QA context as an additional node (*QA context node*) and connect it to the topic entities in the KG subgraph as shown in Figure 1. This joint graph, which we term the *working graph*, unifies the two modalities into one graph. We then augment the feature of each node with the relevance score, and design a new attention-based GNN module for reasoning. Our joint reasoning algorithm on the working graph simultaneously updates the representation of both the KG entities and the QA context node, bridging the gap between the two sources of information.

We evaluate QA-GNN on two question answering datasets that require reasoning with knowledge: *CommonsenseQA* (Talmor et al., 2019) and *OpenBookQA* (Mihaylov et al., 2018), using the *ConceptNet* KG (Speer et al., 2017). QA-GNN outperforms strong fine-tuned LM baselines as well as the existing best LM+KG model (with the same LM) by up to 5.7% and 3.7% respectively. In particular, QA-GNN exhibits improved performance on some forms of structured reasoning (*e.g.*, correctly handling negation and entity substitution in questions): it achieves 4.6% improvement over fine-tuned LMs on questions with negation, while existing LM+KG models are +0.6% over fine-tuned LMs. We also show that one can extract reasoning processes from QA-GNN in the form of general KG subgraphs, not just paths (Lin et al., 2019), suggesting a general method for explaining model predictions.

2 Problem Statement

We aim to answer natural language questions using knowledge from a pre-trained LM and a structured KG. We use the term language model broadly to be any composition of two functions, $f_{\text{head}}(f_{\text{enc}}(\mathbf{x}))$, where f_{enc}, the encoder, maps a textual input $\mathbf{x}$ to a contextualized vector representation $\mathbf{h}^{\text{LM}}$, and f_{head} uses this representation to perform a desired task (which we discuss in §3.2). In this work, we specifically use masked language models (*e.g.*, RoBERTa) as f_{enc}, and let $\mathbf{h}^{\text{LM}}$ denote the output representation of a `[CLS]` token that is prepended to the input sequence $\mathbf{x}$, unless otherwise noted. We define the knowledge graph as a multi-relational graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Here $\mathcal{V}$ is the set of entity nodes in the KG; $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{R} \times \mathcal{V}$ is the set of edges that connect nodes in $\mathcal{V}$, where $\mathcal{R}$ represents a set of relation types.

Given a question q and an answer choice $a \in \mathcal{C}$, we follow prior work (Lin et al., 2019) to link the entities mentioned in the question and answer choice to the given KG $\mathcal{G}$. We denote $\mathcal{V}_q \subseteq \mathcal{V}$ and $\mathcal{V}_a \subseteq \mathcal{V}$ as the set of KG entities mentioned in the question (*question entities*; **blue** entities in Figure 1) and answer choice (*answer choice entities*; **red** entities in Figure 1), respectively, and use $\mathcal{V}_{q,a} := \mathcal{V}_q \cup \mathcal{V}_a$ to denote all the entities that appear in either the question or answer choice, which we call *topic entities*. We then extract a subgraph from $\mathcal{G}$ for a question-choice pair, $\mathcal{G}^{q,a}_{\text{sub}} = (\mathcal{V}^{q,a}_{\text{sub}}, \mathcal{E}^{q,a}_{\text{sub}})$,[1] which comprises all nodes on the k-hop paths between nodes in $\mathcal{V}_{q,a}$.

[1] We remove the superscript q,a if there is no ambiguity.

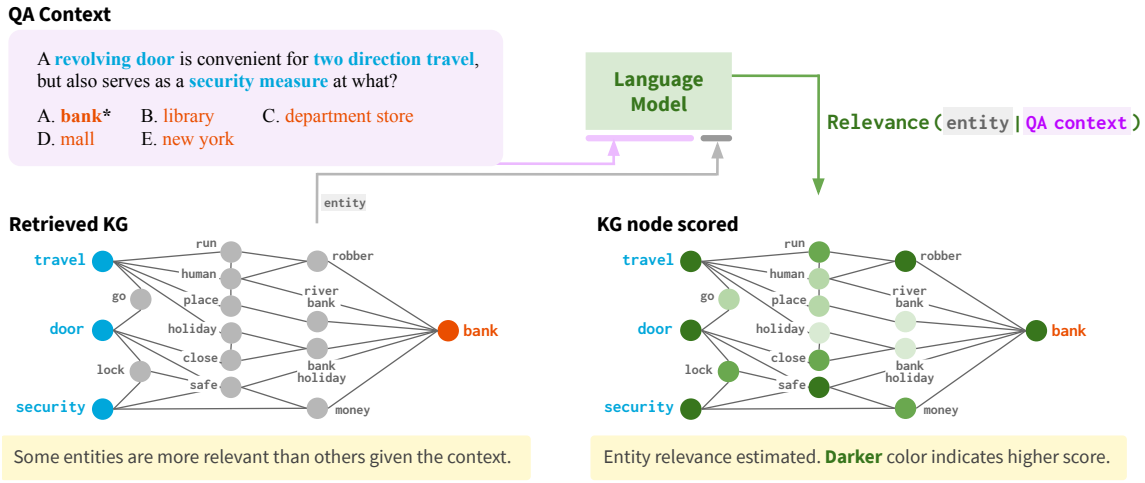

Figure 3: Relevance scoring of the retrieved KG: we use a pre-trained LM to calculate the relevance of each KG entity node conditioned on the QA context (§3.2).

3 Approach: QA-GNN

As shown in Figure 2, given a question and an answer choice a, we concatenate them to get the *QA context* $[q; a]$. To reason over a given QA context using knowledge from both the LM and the KG, QA-GNN works as follows. First, we use the LM to obtain a representation for the QA context, and retrieve the subgraph $\mathcal{G}_{\text{sub}}$ from the KG. Then we introduce a *QA context node* z that represents the QA context, and connect z to the topic entities $\mathcal{V}_{q,a}$ so that we have a joint graph over the two sources of knowledge, which we term the *working graph*, $\mathcal{G}_{\text{W}}$ (§3.1). To adaptively capture the relationship between the QA context node and each of the other nodes in $\mathcal{G}_{\text{W}}$, we calculate a relevance score for each pair using the LM, and use this score as an additional feature for each node (§3.2). We then propose an attention-based GNN module that does message passing on the $\mathcal{G}_{\text{W}}$ for multiple rounds (§3.3). Finally, we make the final prediction using the LM representation, QA context node representation and a pooled working graph representation (§3.4).

3.1 Joint graph representation

To design a joint reasoning space for the two sources of knowledge, we explicitly connect them in a common graph structure. We introduce a new QA context node z which represents the QA context, and connect z to each topic entity in $\mathcal{V}_{q,a}$ on the KG subgraph $\mathcal{G}_{\text{sub}}$ using two new relation types $r_{z,q}$ and $r_{z,a}$. These relation types capture the relationship between the QA context and the relevant entities in the KG, depending on whether the entity is found in the question portion or the answer portion of the QA context. Since this joint graph intuitively provides a reasoning space (working memory) over the QA context and KG, we term it ***working graph*** $\mathcal{G}_{\text{W}} = (\mathcal{V}_{\text{W}}, \mathcal{E}_{\text{W}})$, where $\mathcal{V}_{\text{W}} = \mathcal{V}_{\text{sub}} \cup \{z\}$ and $\mathcal{E}_{\text{W}} = \mathcal{E}_{\text{sub}} \cup \{(z, r_{z,q}, v) \,|\, v \in \mathcal{V}_q\} \cup \{(z, r_{z,a}, v) \,|\, v \in \mathcal{V}_a\}$.

Each node in the $\mathcal{G}_{\text{W}}$ is associated with one of the four types: $\mathcal{T} = \{\mathbf{Z}, \mathbf{Q}, \mathbf{A}, \mathbf{O}\}$, each indicating the context node z, nodes in $\mathcal{V}_q$, nodes in $\mathcal{V}_a$, and other nodes, respectively (corresponding to the node color, **purple**, **blue**, **red**, gray in Figure 1 and 2). We denote the text of the context node z (QA context) and KG node $v \in \mathcal{V}_{\text{sub}}$ (entity name) as $\texttt{text}(z)$ and $\texttt{text}(v)$.

We initialize the node embedding for z using the LM representation of the QA context ($\boldsymbol{z}^{\text{LM}} = f_{\text{enc}}(\texttt{text}(z))$), and each node on the $\mathcal{G}_{\text{sub}}$ using the entity embedding from Feng et al. (2020). In the subsequent sections, we will reason over the working graph in order to score a given (question, answer choice) pair.

3.2 KG node relevance scoring

Many nodes on the KG subgraph $\mathcal{G}_{\text{sub}}$ (*i.e.*, those heuristically retrieved from the KG) can be irrelevant under the current QA context. As an example shown in Figure 3, the retrieved KG subgraph $\mathcal{G}_{\text{sub}}$ with few-hop neighbors of the $\mathcal{V}_{q,a}$ may include nodes that are uninformative for the reasoning process, *e.g.*, nodes "holiday" and "river bank" are off-topic; "human" and "place" are generic. These irrelevant nodes may result in overfitting or introduce unnecessary difficulty in reasoning, an issue especially when $\mathcal{V}_{q,a}$ is large. For instance, we empirically find that using the *ConceptNet* KG (Speer et al., 2017), we will retrieve a KG with $|\mathcal{V}_{\text{sub}}| > 400$ nodes on average if we consider 3-hop neighbors.

In response, we propose node relevance scoring, where we use the pre-trained language model to score the relevance of each KG node $v \in \mathcal{V}_{\text{sub}}$

conditioned on the QA context. For each node v, we concatenate the entity $\texttt{text}(v)$ with the QA context $\texttt{text}(z)$ and compute the *relevance score*:

$$\rho_v = f_{\text{head}}(f_{\text{enc}}([\texttt{text}(z); \texttt{text}(v)])), \quad (1)$$

where $f_{\text{head}} \circ f_{\text{enc}}$ denotes the probability of $\texttt{text}(v)$ computed by the LM. This relevance score ρ_v captures the importance of each KG node relative to the given QA context, which is used for reasoning or pruning the working graph $\mathcal{G}_{\text{W}}$.

3.3 GNN architecture

To perform reasoning on the working graph $\mathcal{G}_{\text{W}}$, our GNN module builds on the graph attention framework (GAT) (Veličković et al., 2018), which induces node representations via iterative message passing between neighbors on the graph. Specifically, in a L-layer QA-GNN, for each layer, we update the representation $\boldsymbol{h}_t^{(\ell)} \in \mathbb{R}^D$ of each node $t \in \mathcal{V}_{\text{W}}$ by

$$\boldsymbol{h}_t^{(\ell+1)} = f_n\left(\sum_{s \in \mathcal{N}_t \cup \{t\}} \alpha_{st} \boldsymbol{m}_{st}\right) + \boldsymbol{h}_t^{(\ell)}, \quad (2)$$

where $\mathcal{N}_t$ represents the neighborhood of node t, $\boldsymbol{m}_{st} \in \mathbb{R}^D$ notes the message from each neighbor node s to t, and α_{st} is an attention weight that scales each message $\boldsymbol{m}_{st}$ from s to t. The sum of the messages is then passed through a 2-layer MLP, f_n: $\mathbb{R}^D \to \mathbb{R}^D$, with batch normalization (Ioffe and Szegedy, 2015). For each node $t \in \mathcal{V}_{\text{W}}$, we set $\boldsymbol{h}_t^{(0)}$ using a linear transformation f_h that maps its initial node embedding (described in §3.1) to $\mathbb{R}^D$. Crucially, as our GNN message passing operates on the working graph, it will jointly leverage and update the representation of the QA context and KG. We further propose an expressive message ($\boldsymbol{m}_{st}$) and attention (α_{st}) computation below.

Node type & relation-aware message. As $\mathcal{G}_{\text{W}}$ is a multi-relational graph, the message passed from a source node to the target node should capture their relationship, *i.e.*, relation type of the edge and source/target node types. To this end, we first obtain the type embedding $\boldsymbol{u}_t$ of each node t, as well as the relation embedding $\boldsymbol{r}_{st}$ from node s to node t by

$$\boldsymbol{u}_t = f_u(\mathtt{u}_t), \quad \boldsymbol{r}_{st} = f_r(\mathtt{e}_{st}, \mathtt{u}_s, \mathtt{u}_t), \quad (3)$$

where $\mathtt{u}_s, \mathtt{u}_t \in \{0,1\}^{|\mathcal{T}|}$ are one-hot vectors indicating the node types of s and t, $\mathtt{e}_{st} \in \{0,1\}^{|\mathcal{R}|}$ is a one-hot vector indicating the relation type of edge (s,t), f_u: $\mathbb{R}^{|\mathcal{T}|} \to \mathbb{R}^{D/2}$ is a linear transformation, and f_r: $\mathbb{R}^{|\mathcal{R}|+2|\mathcal{T}|} \to \mathbb{R}^D$ is a 2-layer MLP. We then compute the message from s to t as

$$\boldsymbol{m}_{st} = f_m(\boldsymbol{h}_s^{(\ell)}, \boldsymbol{u}_s, \boldsymbol{r}_{st}), \quad (4)$$

where f_m: $\mathbb{R}^{2.5D} \to \mathbb{R}^D$ is a linear transformation.

Node type, relation, and score-aware attention. Attention captures the strength of association between two nodes, which is ideally informed by their node types, relations and node relevance scores. We first embed the relevance score of each node t by

$$\boldsymbol{\rho}_t = f_\rho(\rho_t), \quad (5)$$

where f_ρ: $\mathbb{R} \to \mathbb{R}^{D/2}$ is an MLP. To compute the attention weight α_{st} from node s to node t, we obtain the query and key vectors $\boldsymbol{q}$, $\boldsymbol{k}$ by

$$\boldsymbol{q}_s = f_q(\boldsymbol{h}_s^{(\ell)}, \boldsymbol{u}_s, \boldsymbol{\rho}_s), \quad (6)$$

$$\boldsymbol{k}_t = f_k(\boldsymbol{h}_t^{(\ell)}, \boldsymbol{u}_t, \boldsymbol{\rho}_t, \boldsymbol{r}_{st}), \quad (7)$$

where f_q: $\mathbb{R}^{2D} \to \mathbb{R}^D$ and f_k: $\mathbb{R}^{3D} \to \mathbb{R}^D$ are linear transformations. The attention weight is then

$$\alpha_{st} = \frac{\exp(\gamma_{st})}{\sum_{t' \in \mathcal{N}_s \cup \{s\}} \exp(\gamma_{st'})}, \quad \gamma_{st} = \frac{\boldsymbol{q}_s^\top \boldsymbol{k}_t}{\sqrt{D}}. \quad (8)$$

3.4 Inference & Learning

Given a question q and an answer choice a, we use the information from both the QA context and the KG to calculate the probability of it being the answer $p(a \mid q) \propto \exp(\text{MLP}(\boldsymbol{z}^{\text{LM}}, \boldsymbol{z}^{\text{GNN}}, \boldsymbol{g}))$, where $\boldsymbol{z}^{\text{GNN}} = \boldsymbol{h}_z^{(L)}$ and $\boldsymbol{g}$ denotes the pooling of $\{\boldsymbol{h}_v^{(L)} \mid v \in \mathcal{V}_{\text{sub}}\}$. In the training data, each question has a set of answer choices with one correct choice. We optimize the model (both the LM and GNN components end-to-end) using the cross entropy loss.

3.5 Computation complexity

We analyze the time and space complexity of our method and compare with prior works, KagNet (Lin et al., 2019) and MHGRN (Feng et al., 2020) in Table 1. As we handle edges of different relation types using different edge embeddings instead of designing an independent graph networks for each relation as in RGCN (Schlichtkrull et al., 2018) or MHGRN, the time complexity of our method is constant with respect to the number of relations and linear with respect to the number of nodes. We achieve the same space complexity as MHGRN (Feng et al., 2020).

4 Experiments

4.1 Datasets

We evaluate QA-GNN on two question answering datasets: *CommonsenseQA* (Talmor et al., 2019) and *OpenBookQA* (Mihaylov et al., 2018). **CommonsenseQA** is a 5-way multiple choice QA task that requires reasoning with commonsense knowledge, containing 12,102 questions. The test set of CommonsenseQA is not publicly available, and model predictions can only be evaluated once every two weeks via the official leaderboard. Hence,

Model	Time	Space
$\mathcal{G}$ is a dense graph		
L-hop KagNet	$\mathcal{O}(\lvert\mathcal{R}\rvert^L\lvert\mathcal{V}\rvert^{L+1}L)$	$\mathcal{O}(\lvert\mathcal{R}\rvert^L\lvert\mathcal{V}\rvert^{L+1}L)$
L-hop MHGRN	$\mathcal{O}(\lvert\mathcal{R}\rvert^2\lvert\mathcal{V}\rvert^2L)$	$\mathcal{O}(\lvert\mathcal{R}\rvert\lvert\mathcal{V}\rvert L)$
L-layer QA-GNN	$\mathcal{O}(\lvert\mathcal{V}\rvert^2L)$	$\mathcal{O}(\lvert\mathcal{R}\rvert\lvert\mathcal{V}\rvert L)$
$\mathcal{G}$ is a sparse graph with maximum node degree $\Delta \ll \lvert\mathcal{V}\rvert$		
L-hop KagNet	$\mathcal{O}(\lvert\mathcal{R}\rvert^L\lvert\mathcal{V}\rvert L\Delta^L)$	$\mathcal{O}(\lvert\mathcal{R}\rvert^L\lvert\mathcal{V}\rvert L\Delta^L)$
L-hop MHGRN	$\mathcal{O}(\lvert\mathcal{R}\rvert^2\lvert\mathcal{V}\rvert L\Delta)$	$\mathcal{O}(\lvert\mathcal{R}\rvert\lvert\mathcal{V}\rvert L)$
L-layer QA-GNN	$\mathcal{O}(\lvert\mathcal{V}\rvert L\Delta)$	$\mathcal{O}(\lvert\mathcal{R}\rvert\lvert\mathcal{V}\rvert L)$

Table 1: **Computation complexity** of different L-hop reasoning models on a dense/sparse graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the relation set $\mathcal{R}$.

we perform main experiments on the **in-house (IH) data split** used in Lin et al. (2019), and also report the score of our final system on the **official test** set. **OpenBookQA** is a 4-way multiple choice QA task that requires reasoning with elementary science knowledge, containing 5,957 questions. We use the official data split.

4.2 Knowledge graphs

We use *ConceptNet* (Speer et al., 2017), a general-domain knowledge graph, as our structured knowledge source $\mathcal{G}$ for both of the above tasks. Given each QA context (question and answer choice), we retrieve the subgraph $\mathcal{G}_{\text{sub}}$ from $\mathcal{G}$ following the pre-processing step described in Feng et al. (2020), with hop size $k = 2$. Henceforth, in this section (§4) we use the term "KG" to refer to $\mathcal{G}_{\text{sub}}$.

4.3 Implementation & training details

We set the dimension ($D = 200$) and number of layers ($L = 5$) of our GNN module, with dropout rate 0.2 applied to each layer (Srivastava et al., 2014). The parameters of the model are optimized by RAdam (Liu et al., 2020), with batch size 128, gradient clipping 1.0 (Pascanu et al., 2013), and learning rate 1e-5 and 1e-3 for the LM and GNN components respectively. Each model is trained using two GPUs (GTX Titan X), which takes ~20 hours on average. The above hyperparameters were tuned on the development set.

4.4 Baselines

Fine-tuned LM. To study the role of KGs, we compare with a vanilla fine-tuned LM, which does not use the KG. We use RoBERTa-large (Liu et al., 2019) for *CommonsenseQA*, and RoBERTa-large and AristoRoBERTa[2] (Clark et al., 2019) for *OpenBookQA*.

[2]OpenBookQA provides an extra corpus of scientific facts in a textual form. AristoRoBERTa uses the facts corresponding to each question, prepared by Clark et al. (2019), as an additional input to the QA context.

Methods	IHdev-Acc. (%)	IHtest-Acc. (%)
RoBERTa-large (w/o KG)	73.07 (±0.45)	68.69 (±0.56)
+ RGCN (Schlichtkrull et al., 2018)	72.69 (±0.19)	68.41 (±0.66)
+ GconAttn (Wang et al., 2019a)	72.61(±0.39)	68.59 (±0.96)
+ 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)
+ MHGRN (Feng et al., 2020)	74.45 (±0.10)	71.11 (±0.81)
+ QA-GNN (**Ours**)	**76.54** (±0.21)	**73.41** (±0.92)

Table 2: **Performance comparison on *Commonsense QA* in-house split** (controlled experiments). As the official test is hidden, here we report the in-house Dev (IHdev) and Test (IHtest) accuracy, following the data split of Lin et al. (2019).

Methods	Test
RoBERTa (Liu et al., 2019)	72.1
RoBERTa+FreeLB (Zhu et al., 2020) (ensemble)	73.1
RoBERTa+HyKAS (Ma et al., 2019)	73.2
RoBERTa+KE (ensemble)	73.3
RoBERTa+KEDGN (ensemble)	74.4
XLNet+GraphReason (Lv et al., 2020)	75.3
RoBERTa+MHGRN (Feng et al., 2020)	75.4
Albert+PG (Wang et al., 2020b)	75.6
Albert (Lan et al., 2020) (ensemble)	76.5
UnifiedQA* (Khashabi et al., 2020)	**79.1**
RoBERTa + QA-GNN (**Ours**)	76.1

Table 3: **Test accuracy on *CommonsenseQA*'s official leaderboard**. The top system, UnifiedQA (11B parameters) is 30x larger than our model.

Existing LM+KG models. We compare with existing LM+KG methods, which share the same high-level framework as ours but use different modules to reason on the KG in place of QA-GNN ("yellow box" in Figure2): (1) Relation Network (RN) (Santoro et al., 2017), (2) RGCN (Schlichtkrull et al., 2018), (3) GconAttn (Wang et al., 2019a), (4) KagNet (Lin et al., 2019), and (5) MHGRN (Feng et al., 2020). (1),(2),(3) are relation-aware GNNs for KGs, and (4),(5) further model paths in KGs. MHGRN is the existing top performance model under this LM+KG framework. For fair comparison, we use the same LM in all the baselines and our model. The key differences between QA-GNN and these are that they do not perform relevance scoring or joint updates with the QA context (§3).

4.5 Main results

Table 2 and Table 4 show the results on CommonsenseQA and OpenBookQA, respectively. On both datasets, we observe consistent improvements over fine-tuned LMs and existing LM+KG models, *e.g.*, on OpenBookQA, +5.7% over RoBERTa, and +3.7% over the prior best LM+KG system,

Methods	RoBERTa-large	AristoRoBERTa
Fine-tuned LMs (w/o KG)	64.80 (±2.37)	78.40 (±1.64)
+ RGCN	62.45 (±1.57)	74.60 (±2.53)
+ GconAtten	64.75 (±1.48)	71.80 (±1.21)
+ RN	65.20 (±1.18)	75.35 (±1.39)
+ MHGRN	66.85 (±1.19)	80.6
+ QA-GNN (**Ours**)	**70.58** (±1.42)	**82.77** (±1.56)

Table 4: **Test accuracy comparison on *OpenBook QA*** (controlled experiments). Methods with AristoRoBERTa use the textual evidence by Clark et al. (2019) as an additional input to the QA context.

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

Table 5: **Test accuracy on *OpenBookQA* leaderboard**. All listed methods use the provided science facts as an additional input to the language context. The top 2 systems, UnifiedQA (11B params) and T5 (3B params) are 30x and 8x larger than our model.

Graph Connection (§3.1)	Dev Acc.
No edge between Z and KG nodes	74.81
Connect Z to all KG nodes	76.38
Connect Z to QA entity nodes (**final**)	**76.54**

Relevance scoring (§3.2)	Dev Acc.
Nothing	75.56
w/ contextual embedding	76.31
w/ relevance score (**final**)	**76.54**
w/ both	76.52

GNN Attention & Message (§3.3)	Dev Acc.
Node type, relation, score-aware (**final**)	**76.54**
- type-aware	75.41
- relation-aware	75.61
- score-aware	75.56

GNN Layers (§3.3)	Dev Acc.
$L = 3$	75.53
$L = 4$	76.34
$L = 5$ (**final**)	**76.54**
$L = 6$	76.21
$L = 7$	75.96

Table 6: **Ablation study** of our model components, using the CommonsenseQA IHdev set.

MHGRN. The boost over MHGRN suggests that QA-GNN makes a better use of KGs to perform joint reasoning than existing LM+KG methods.

We also achieve competitive results to other systems on the official leaderboards (Table 3 and 5). Notably, the top two systems, T5 (Raffel et al., 2020) and UnifiedQA (Khashabi et al., 2020), are trained with more data and use 8x to 30x more parameters than our model (ours has ∼360M parameters). Excluding these and ensemble systems, our model is comparable in size and amount of data to other systems, and achieves the top performance on the two datasets.

4.6 Analysis

4.6.1 Ablation studies

Table 6 summarizes the ablation study conducted on each of our model components (§3.1, §3.2, §3.3), using the CommonsenseQA IHdev set.

Graph connection (top left table): The first key component of QA-GNN is the joint graph that connects the z node (QA context) to QA entity nodes $\mathcal{V}_{q,a}$ in the KG (§3.1). Without these edges, the QA context and KG cannot mutually update their representations, hurting the performance: 76.5% →74.8%, which is close to the previous LM+KG system, MHGRN. If we connected z to all the nodes in the KG (not just QA entities), the performance is comparable or drops slightly (-0.16%).

KG node relevance scoring (top right table): We find the relevance scoring of KG nodes (§3.2) provides a boost: 75.56% → 76.54%. As a variant of the relevance scoring in Eq. 1, we also experimented with obtaining a *contextual embedding* $\boldsymbol{w}_v$ for each node $v \in \mathcal{V}_{\text{sub}}$ and adding to the node features: $\boldsymbol{w}_v = f_{\text{enc}}([\texttt{text}(z); \texttt{text}(v)])$. However, we find that it does not perform as well (76.31%), and using both the relevance score and contextual embedding performs on par with using the score alone, suggesting that the score has a sufficient information in our tasks; hence, our final system simply uses the relevance score.

GNN architecture (bottom tables): We ablate the information of node type, relation, and relevance score from the attention and message computation in the GNN (§3.3). The results suggest that all these features improve the model performance. For the number of GNN layers, we find $L = 5$ works the best on the dev set. Our intuition is that 5 layers allow various message passing or reasoning patterns between the QA context (z) and KG, such as "$z \rightarrow$ 3 hops on KG nodes $\rightarrow z$".

4.6.2 Model interpretability

We aim to interpret QA-GNN's reasoning process by analyzing the node-to-node attention weights induced by the GNN. Figure 4 shows two examples. In (a), we perform Best First Search (BFS) on the working graph to trace high attention weights from the QA context node (**Z**; purple) to **Q**uestion entity nodes (blue) to **O**ther (gray) or **A**nswer choice entity nodes (orange), which reveals that the QA context z attends to "elevator" and "basement" in the KG, "elevator" and "basement" both attend strongly to "building", and "building" attends to "office building", which is our final answer. In (b),

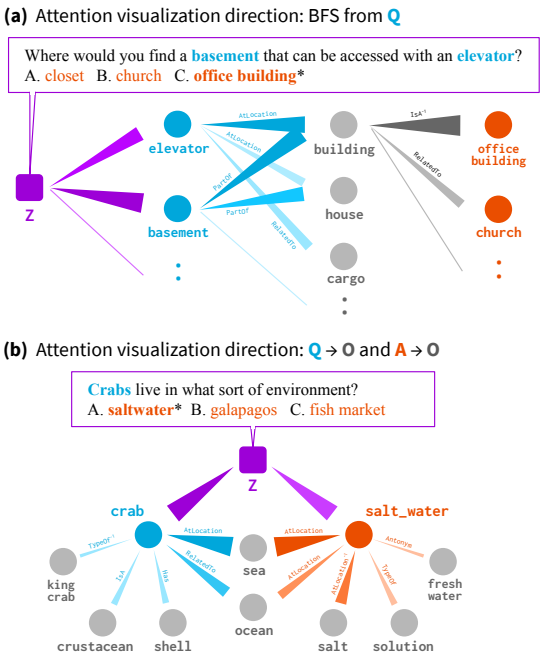

Figure 4: **Interpreting QA-GNN's reasoning process** by analyzing the node-to-node attention weights induced by the GNN. Darker and thicker edges indicate higher attention weights.

we use BFS to trace attention weights from two directions: $\mathbf{Z} \rightarrow \mathbf{Q} \rightarrow \mathbf{O}$ and $\mathbf{Z} \rightarrow \mathbf{A} \rightarrow \mathbf{O}$, which reveals concepts ("sea" and "ocean") in the KG that are not necessarily mentioned in the QA context but bridge the reasoning between the question entity ("crab") and answer choice entity ("salt water"). While prior KG reasoning models (Lin et al., 2019; Feng et al., 2020) enumerate individual paths in the KG for model interpretation, QA-GNN is not specific to paths, and helps to find more general reasoning structures (*e.g.*, a KG subgraph with multiple anchor nodes as in example (a)).

4.6.3 Structured reasoning

Structured reasoning, *e.g.*, precise handling of negation or entity substitution (*e.g.*, "hair" → "art" in Figure 5b) in question, is crucial for making robust predictions. Here we analyze QA-GNN's ability to perform structured reasoning and compare with baselines (fine-tuned LMs and existing LM+KG models).

Quantitative analysis. Table 7 compares model performance on questions containing negation words (*e.g.*, no, not, nothing, unlikely), taken from the CommonsenseQA IHtest set. We find that previous LM+KG models (KagNet, MHGRN) provide limited improvements over RoBERTa on questions with negation (+0.6%); whereas QA-GNN exhibits a bigger boost (+4.6%), suggesting its strength

Methods	IHtest-Acc. (Overall)	IHtest-Acc. (Question w/ **negation**)
RoBERTa-large (w/o KG)	68.7	54.2
+ KagNet	69.0 (+0.3)	54.2 (+0.0)
+ MHGRN	71.1 (+2.4)	54.8 (+0.6)
+ QA-GNN (**Ours**)	73.4 (+4.7)	**58.8 (+4.6)**
+ QA-GNN (no edge between Z and KG)	71.5 (+2.8)	55.1 (+0.9)

Table 7: Performance on **questions with negation** in *CommonsenseQA*. () shows the difference with RoBERTa. Existing LM+KG methods (KagNet, MHGRN) provide limited improvements over RoBERTa (+0.6%); QA-GNN exhibits a bigger boost (+4.6%), suggesting its strength in structured reasoning.

in structured reasoning. We hypothesize that QA-GNN's joint updates of the representations of the QA context and KG (during GNN message passing) allows the model to integrate semantic nuances expressed in language. To further study this hypothesis, we remove the connections between z and KG nodes from our QA-GNN (Table 7 bottom): now the performance on negation becomes close to the prior work, MHGRN, suggesting that the joint message passing helps for performing structured reasoning.

Qualitative analysis. Figure 5 shows a case study to analyze our model's behavior for structured reasoning. The question on the left contains negation "**not** used for hair", and the correct answer is "B. art supply". We observe that in the 1st layer of QA-GNN, the attention from z to question entities ("hair", "round brush") is diffuse. After multiples rounds of message passing on the working graph, z attends strongly to "round brush" in the final layer of the GNN, but weakly to the negated entity "hair". The model correctly predicts the answer "B. art supply". Next, given the original question on the left, we (a) drop the negation or (b) modify the topic entity ("hair" → "art"). In (a), z now attends strongly to "hair", which is not negated anymore. The model predicts the correct answer "A. hair brush". In (b), we observe that QA-GNN recognizes the same structure as the original question (with only the entity swapped): z attends weakly to the negated entity ("art") like before, and the model correctly predicts "A. hair brush" over "B. art supply".

Table 8 shows additional examples, where we compare QA-GNN's predictions with the LM baseline (RoBERTa). We observe that RoBERTa tends to make the same prediction despite the modifications we make to the original questions (*e.g.*, drop/insert negation, change an entity); on the other hand, QA-GNN adapts predictions to the modifications correctly (except for double negation

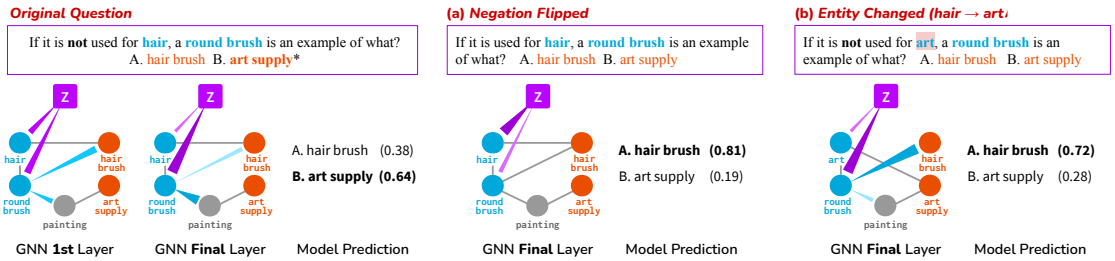

Figure 5: **Analysis of QA-GNN's behavior for structured reasoning**. Given an original question (left), we modify its negation (middle) or topic entity (right): we find that QA-GNN adapts attention weights and final predictions accordingly, suggesting its capability to handle structured reasoning.

Example (Original taken from *CommonsenseQA* Dev)	**RoBERTa Prediction**	**Our Prediction**
[Original] If it is **not** used for hair, a round brush is an example of what? A. hair brush B. art supply	A. hair brush (✗)	B. art supply (✓)
[Negation flip] If it is used for hair, a round brush is an example of what?	A. hair brush (✓ just no change?)	A. hair brush (✓)
[Entity change] If it is not used for art, a round brush is an example of what?	A. hair brush (✓ just no change?)	A. hair brush (✓)
[Original] If you have to read a book that is very dry you may become what? A. interested B. bored	B. bored (✓)	B. bored (✓)
[Negation ver 1] If you have to read a book that is very dry you may **not** become what?	B. bored (✗)	A. interested (✓)
[Negation ver 2] If you have to read a book that is **not** dry you may become what?	B. bored (✗)	A. interested (✓)
[Double negation] If you have to read a book that is **not** dry you may **not** become what?	B. bored (✓ just no change?)	A. interested (✗)

Table 8: **Case study of structured reasoning**, comparing predictions by RoBERTa and our model (RoBERTa + QA-GNN). Our model correctly handles changes in negation and topic entities.

Methods	**IHtest-Acc.** (Question w/ ≤10 entities)	**IHtest-Acc.** (Question w/ >10 entities)
RoBERTa-large (w/o KG)	68.4	70.0
+ MHGRN	71.5	70.1
+ QA-GNN (w/o node relevance score)	72.8 (+1.3)	71.5 (+1.4)
+ QA-GNN (w/ node relevance score; **final system**)	73.4 (+1.9)	**73.5 (+3.4)**

Table 9: Performance on **questions with fewer/more entities** in *CommonsenseQA*. () shows the difference with MHGRN (LM+KG baseline). KG node relevance scoring (§3.2) boosts the performance on questions containing more entities (i.e. larger retrieved KG).

in the table bottom, which is a future work).

4.6.4 Effect of KG node relevance scoring

We find that KG node relevance scoring (§3.2) is helpful when the retrieved KG ($\mathcal{G}_{\text{sub}}$) is large. Table 9 shows model performance on questions containing fewer (≤10) or more (>10) entities in the CommonsenseQA IHtest set (on average, the former and latter result in 90 and 160 nodes in $\mathcal{G}_{\text{sub}}$, respectively). Existing LM+KG models such as MHGRN achieve limited performance on questions with more entities due to the size and noisiness of retrieved KGs: 70.1% accuracy vs 71.5% accuracy on questions with fewer entities. KG node relevance scoring mitigates this bottleneck, reducing the accuracy discrepancy: 73.5% and 73.4% accuracy on questions with more/fewer entities, respectively.

5 Related work and discussion

Knowledge-aware methods for NLP. Various works have studied methods to augment NLP systems with knowledge. Existing works (Pan et al., 2019; Ye et al., 2019; Petroni et al., 2019; Bosselut et al., 2019) study pre-trained LMs' potential as latent knowledge bases. To provide more explicit and interpretable knowledge, several works integrate structured knowledge (KGs) into LMs (Mihaylov and Frank, 2018; Lin et al., 2019; Wang et al., 2019a; Yang et al., 2019; Wang et al., 2020b; Bosselut et al., 2021).

Question answering with LM+KG. In particular, a line of works propose LM+KG methods for question answering. Most closely related to ours are works by Lin et al. (2019); Feng et al. (2020); Lv et al. (2020). Our novelties are (1) the joint graph of QA context and KG, on which we *mutually* update the representations of the LM and KG; and (2) *language-conditioned* KG node relevance scoring. Other works on scoring or pruning KG nodes/paths rely on graph-based metrics such as PageRank, centrality, and off-the-shelf KG embeddings (Paul and Frank, 2019; Fadnis et al., 2019; Bauer et al., 2018; Lin et al., 2019), without reflecting the QA context.

Other QA tasks. Several works study other forms of question answering tasks, *e.g.*, passage-based QA, where systems identify answers using given or retrieved documents (Rajpurkar et al., 2016; Joshi et al., 2017; Yang et al., 2018), and

KBQA, where systems perform semantic parsing of a given question and execute the parsed queries on knowledge bases (Berant et al., 2013; Yih et al., 2016; Yu et al., 2018). Different from these tasks, we approach question answering using knowledge available in LMs and KGs.

Knowledge representations. Several works study joint representations of external textual knowledge (*e.g.*, Wikipedia articles) and structured knowledge (*e.g.*, KGs) (Riedel et al., 2013; Toutanova et al., 2015; Xiong et al., 2019; Sun et al., 2019; Wang et al., 2019b). The primary distinction of our joint graph representation is that we construct a graph connecting each *question* and KG rather than textual and structural knowledge, approaching a complementary problem to the above works.

Graph neural networks (GNNs). GNNs have been shown to be effective for modeling graph-based data. Several works use GNNs to model the structure of text (Yasunaga et al., 2017; Zhang et al., 2018; Yasunaga and Liang, 2020) or KGs (Wang et al., 2020a). In contrast to these works, QA-GNN jointly models the language and KG. Graph Attention Networks (GATs) (Veličković et al., 2018) perform attention-based message passing to induce graph representations. We build on this framework, and further condition the GNN on the language input by introducing a QA context node (§3.1), KG node relevance scoring (§3.2), and joint update of the KG and language representations (§3.3).

6 Conclusion

We presented QA-GNN, an end-to-end question answering model that leverages LMs and KGs. Our key innovations include (i) **Relevance scoring**, where we compute the relevance of KG nodes conditioned on the given QA context, and (ii) **Joint reasoning** over the QA context and KGs, where we connect the two sources of information via the working graph, and jointly update their representations through GNN message passing. Through both quantitative and qualitative analyses, we showed QA-GNN's improvements over existing LM and LM+KG models on question answering tasks, as well as its capability to perform interpretable and structured reasoning, *e.g.*, correctly handling negation in questions.

Acknowledgment

We thank Rok Sosic, Weihua Hu, Jing Huang, Michele Catasta, members of the SNAP research group, P-Lambda group and Project MOWGLI team, as well as our anonymous reviewers for valuable feedback.

We gratefully acknowledge the support of DARPA under Nos. N660011924033 (MCS); Funai Foundation Fellowship; ARO under Nos. W911NF-16-1-0342 (MURI), W911NF-16-1-0171 (DURIP); NSF under Nos. OAC-1835598 (CINES), OAC-1934578 (HDR), CCF-1918940 (Expeditions), IIS-2030477 (RAPID); Stanford Data Science Initiative, Wu Tsai Neuro-sciences Institute, Chan Zuckerberg Biohub, Amazon, JP-Morgan Chase, Docomo, Hitachi, JD.com, KDDI, NVIDIA, Dell, Toshiba, and United Health Group. Hongyu Ren is supported by Masason Foundation Fellowship and the Apple PhD Fellowship. Jure Leskovec is a Chan Zuckerberg Biohub investigator.

Reproducibility

All code and data are available at `https://github.com/michiyasunaga/qagnn`. Experiments are available at `https://worksheets.codalab.org/worksheets/0xf215deb05edf44a2ac353c711f52a25f`.

References

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

Pratyay Banerjee, Kuntal Kumar Pal, Arindam Mitra, and Chitta Baral. 2019. Careful selection of knowledge to solve open book question answering. In *Association for Computational Linguistics (ACL)*.

Junwei Bao, Nan Duan, Zhao Yan, Ming Zhou, and Tiejun Zhao. 2016. Constraint-based question answering with knowledge graph. In *International Conference on Computational Linguistics (COLING)*.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko.

2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Antoine Bosselut, Ronan Le Bras, and Yejin Choi. 2021. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Çelikyilmaz, and Yejin Choi. 2019. Comet: Commonsense transformers for automatic knowledge graph construction. In *Association for Computational Linguistics (ACL)*.

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

Kshitij Fadnis, Kartik Talamadupula, Pavan Kapanipathi, Haque Ishfaq, Salim Roukos, and Achille Fokoue. 2019. Heuristics for interpretable knowledge graph contextualization. *arXiv preprint arXiv:1911.02085*.

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 *Empirical Methods in Natural Language Processing (EMNLP)*.

Kelvin Guu, John Miller, and Percy Liang. 2015. Traversing knowledge graphs in vector space. *Empirical Methods in Natural Language Processing (EMNLP)*.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning (ICML)*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Association for Computational Linguistics (ACL)*.

Nora Kassner and Hinrich Schütze. 2020. Negated and misprimed probes for pretrained language models: Birds can talk, but cannot fly. In *Association for Computational Linguistics (ACL)*.

Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. Unifiedqa: Crossing format boundaries with a single qa system. In *Findings of EMNLP*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations (ICLR)*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations (ICLR)*.

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

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

Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. 2019. Towards generalizable neuro-symbolic systems for commonsense question answering. *arXiv preprint arXiv:1910.14087*.

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 *Empirical Methods in Natural Language Processing (EMNLP)*.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Association for Computational Linguistics (ACL)*.

Xiaoman Pan, Kai Sun, Dian Yu, Jianshu Chen, Heng Ji, Claire Cardie, and Dong Yu. 2019. Improving question answering with external knowledge. *arXiv preprint arXiv:1902.00993*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *International conference on machine learning (ICML)*, pages 1310–1318.

Debjit Paul and Anette Frank. 2019. Ranking and selecting multi-hop knowledge paths to better predict human needs. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? In *Empirical Methods in Natural Language Processing (EMNLP)*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text

transformer. *Journal of Machine Learning Research (JMLR)*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Hongyu Ren, Weihua Hu, and Jure Leskovec. 2020. Query2box: Reasoning over knowledge graphs in vector space using box embeddings. In *International Conference on Learning Representations (ICLR)*.

Hongyu Ren and Jure Leskovec. 2020. Beta embeddings for multi-hop logical reasoning in knowledge graphs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

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

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research (JMLR)*, 15(1):1929–1958.

Haitian Sun, Tania Bedrax-Weiss, and William W Cohen. 2019. Pullnet: Open domain question answering with iterative retrieval on knowledge bases and text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Kathryn Mazaitis, Ruslan Salakhutdinov, and William W Cohen. 2018. Open domain question answering using early fusion of knowledge bases and text. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *North American Chapter of the Association for Computational Linguistics (NAACL)*.

Kristina Toutanova, Danqi Chen, Patrick Pantel, Hoifung Poon, Pallavi Choudhury, and Michael Gamon. 2015. Representing text for joint embedding of text and knowledge bases. In *Empirical Methods in Natural Language Processing (EMNLP)*.

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

Hongwei Wang, Hongyu Ren, and Jure Leskovec. 2020a. Entity context and relational paths for knowledge graph completion. *arXiv preprint arXiv:2002.06757*.

Peifeng Wang, Nanyun Peng, Pedro Szekely, and Xiang Ren. 2020b. Connecting the dots: A knowledgeable path generator for commonsense question answering. *arXiv preprint arXiv:2005.00691*.

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

Xiaozhi Wang, Tianyu Gao, Zhaocheng Zhu, Zhiyuan Liu, Juanzi Li, and Jian Tang. 2019b. Kepler: A unified model for knowledge embedding and pre-trained language representation. *Transactions of the Association for Computational Linguistics (TACL)*.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete kbs with knowledge-aware reader. In *Association for Computational Linguistics (ACL)*.

An Yang, Quan Wang, Jing Liu, Kai Liu, Yajuan Lyu, Hua Wu, Qiaoqiao She, and Sujian Li. 2019. Enhancing pre-trained language representations with rich knowledge for machine reading comprehension. In *Association for Computational Linguistics (ACL)*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Michihiro Yasunaga and Percy Liang. 2020. Graph-based, self-supervised program repair from diagnostic feedback. In *International Conference on Machine Learning (ICML)*.

Michihiro Yasunaga, Rui Zhang, Kshitijh Meelu, Ayush Pareek, Krishnan Srinivasan, and Dragomir Radev. 2017. Graph-based neural multi-document summarization. In *Conference on Computational Natural Language Learning (CoNLL)*.

Zhi-Xiu Ye, Qian Chen, Wen Wang, and Zhen-Hua Ling. 2019. Align, mask and select: A simple method for incorporating commonsense knowledge into language representation models. *arXiv preprint arXiv:1908.06725*.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Association for Computational Linguistics (ACL)*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Thomas Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for language understanding. In *International Conference on Learning Representations (ICLR)*.