KE-GCL: Knowledge Enhanced Graph Contrastive Learning for Commonsense Question Answering

Lihui Zhang[1] and **Ruifan Li**[1,2*]
[1]School of Artificial Intelligence, Beijing University of Posts and Telecommunications, China
[2]Engineering Research Center of Information Networks, Ministry of Education, China
{elliot_zlh, rfli}@bupt.edu.cn

Abstract

Commonsense question answering (CQA) aims to choose the correct answers for commonsense questions. Most existing works focus on extracting and reasoning over external knowledge graphs (KG). However, the noise in KG prevents these models from learning effective representations. In this paper, we propose a **K**nowledge **E**nhanced **G**raph **C**ontrastive **L**earning model (KE-GCL) by incorporating the contextual descriptions of entities and adopting a graph contrastive learning scheme. Specifically, for QA pairs we represent the knowledge from KG and contextual descriptions. Then, the representations of contextual descriptions as context nodes are inserted into KG, forming the knowledge-enhanced graphs. Moreover, we design a contrastive learning method on graphs. For knowledge-enhanced graphs, we build their augmented views with an adaptive sampling strategy. After that, we reason over graphs to update their representations by scattering edges and aggregating nodes. To further improve GCL, hard graph negatives are chosen based on incorrect answers. Extensive experiments on two benchmark datasets demonstrate the effectiveness of our proposed KE-GCL, which outperforms previous methods consistently[1].

1 Introduction

Commonsense question answering (CQA) is an emerging task in the domain of machine reading comprehension with the long-term goal for evaluating the language understanding of machines. The CQA task aims to choose answers for natural language questions about commonsense. Figure 1 shows an example to illustrate the definition of the CQA task. To solve this task, external knowledge graphs (KGs) of commonsense, such as ConceptNet (Speer et al., 2017) where numerous triplets are provided to represent relations between entities, are used in reasoning for the correct answers.

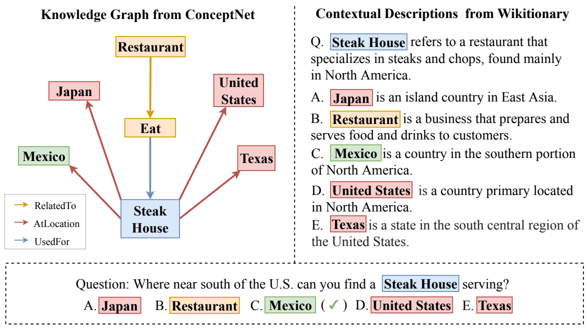

Figure 1: A CQA example from CommonsenseQA dataset. Here, contextual descriptions of entities from Wiktionary are used to enhance the KG from ConceptNet for noise reduction. The entities in red are strong noise.

To take advantage of the commonsense knowledge, a few works (Weissenborn et al., 2017; Santoro et al., 2017; Mihaylov and Frank, 2018; Bauer et al., 2018; Asai et al., 2019) directly retrieve and model the relevant evidence to infer answers. With the great success of graph neural networks (GNNs) (Li et al., 2016; Gilmer et al., 2017; Kipf and Welling, 2017; Schlichtkrull et al., 2018; Veličković et al., 2018; Xu et al., 2019), recent studies (Lin et al., 2019; Qiu et al., 2019; Wang et al., 2019; Xiong et al., 2019; Feng et al., 2020; Lv et al., 2020; Yasunaga et al., 2021) focus on devising exquisite graph networks with task-dependent attention mechanism to model KGs for effective reasoning. However, previous methods ignore the noise, i.e., irrelevant or distracting entities in the KG, resulting in unsatisfactory performance. Take the example in Figure 1. Compared with those between "Steak House" and the other four choices, the relational path between "Steak House" and the choice "Restaurant" is irrelevant to the question. Moreover, the choices with identical relations of "AtLocation" are difficult to be discerned. In other words, the choices whose topology is similar to that

*Corresponding author.
[1]Code and datasets are available at https://github.com/hlhqbzd/KE-GCL.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 76–87

of the correct answer are strong noise. Therefore, the noise problem should be considered seriously.

To address the aforementioned noise problem, we adopt two schemes. **One** is enhancing KG with textual descriptions. As we all know, KG represents the topology among entities. By incorporating contextual meanings, we could further capture the semantic nuances of entities. Thus, it would be beneficial for reducing the negative effect of distracting entities in KGs. Take Figure 1 as an example; with the enhancement of entity descriptions, we undoubtedly exclude distractors (A, D, and E) which are semantically conflict with the location specified in the question ("near south of the U.S."). Then, we derive the correct answer C ("Mexico"). **The other scheme** uses graph contrastive learning (GCL). Intuitively, the KG of the QA pair with the correct answer is often topologically similar to those with distractors. This inherent noise could hinder the graph reasoning. To this end, we construct graph positive pairs with negative counterparts based on the GCL framework.

In this paper, we propose an end-to-end model, **K**nowledge **E**nhanced **G**raph **C**ontrastive **L**earning (KE-GCL) for CQA task. **First**, we concatenate the given QA pair (i.e., a question with its current choice) with the Wiktionary[2] descriptions obtaining the contextual representation. And we extract subgraphs from ConceptNet[3] obtaining the graph embeddings. We then insert the contextual representation as a node into the graph and perform attention-based fusion, obtaining the knowledge-enhanced graph.

Second, we incorporate GCL scheme into our KE-GCL model. To produce the augmented view of the graph, we introduce an adaptive graph augmentation strategy. We adaptively drop out irrelevant edges and mask unimportant node features. The sampling probabilities for nodes and edges are determined by topological connectivity and contextual relevance. Furthermore, to achieve efficient message propagation in graph reasoning, we devise a graph attention network (GAT) (Veličković et al., 2018) based reasoning module, by scattering the connected edges and aggregating the adjacent nodes. The knowledge-enhanced graph and its augmented view jointly perform reasoning. Thus, we obtain the final graph representations.

Third, to enhance the training signals, we build positive and negative pairs for computing graph contrastive loss. Specifically, for the positive pair, we use the graph augmented view of the correct answer. For the negative pairs, we take the graph and its augmented counterparts of other incorrect choices as hard negatives. We set the other graphs in the mini-batch as common negatives. Thus, we train KE-GCL model with a combination of two losses. One is the answer prediction loss and the other is the graph contrastive loss.

Major contributions are summarized as follows:

1) We propose a novel KE-GCL model with a GCL scheme for CQA task. The augmented graph view is generated by adaptively sampling strategy. Hard negatives are chosen based on incorrect answers.

2) We present enhancing the KG with contextual descriptions of entities. A knowledge-enhanced graph is built based on contextual representation. Graph representations are effectively updated via scattering edges and aggregating nodes.

3) We conduct extensive experiments on two benchmark datasets. Experimental results showcase that our KE-GCL achieves better performance compared to the strong baselines consistently.

2 Related Works

KG-aware Methods for CQA. CQA task requires strong capability of knowledge utilization and graph reasoning. Earlier works (Weissenborn et al., 2017; Santoro et al., 2017; Mihaylov and Frank, 2018; Bauer et al., 2018; Asai et al., 2019) inclined to retrieve the reasoning paths between the question and choice entities in the KGs. However, these works were short of knowledge coverage. Recent studies (Lin et al., 2019; Qiu et al., 2019; Wang et al., 2019; Xiong et al., 2019; Feng et al., 2020; Lv et al., 2020; Yasunaga et al., 2021) devoted to utilizing GNNs to encode KGs and aggregate messages from nodes for effective graph reasoning. For instance, Feng et al. (2020) extended the message passing of Relational GCN (Schlichtkrull et al., 2018) to improve the interpretability and scalability of graphs. Lv et al. (2020) retrieved the evidence from ConceptNet and Wikipedia, and constructed heterogeneous graphs for these sources to perform graph-based inference. Yasunaga et al. (2021) used pretrained language models (PLMs) to calculate the relevance between the KG nodes and the QA context, and then performed joint reasoning. These methods over-emphasized the effect of GNNs, but

[2] https://www.wiktionary.org/

[3] https://github.com/commonsense/conceptnet5

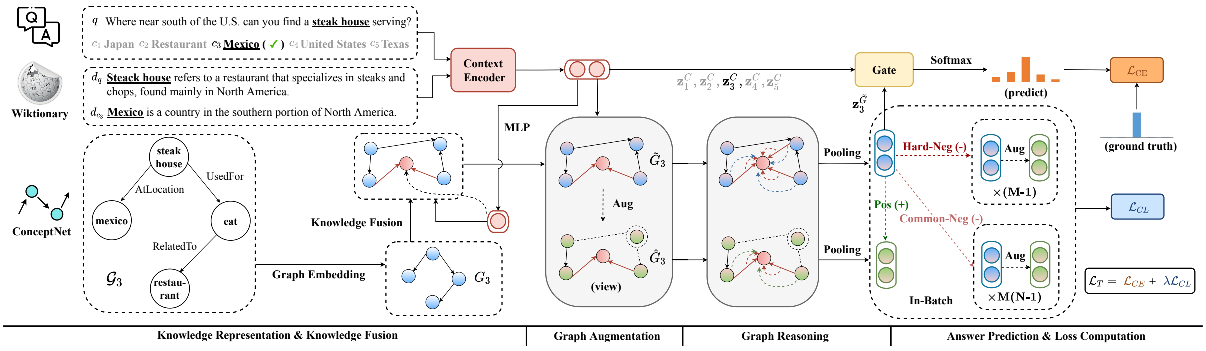

Figure 2: The framework of our KE-GCL model. 1) We represent the QA pair (q, c_3) with its Wiktionary descriptions (d_q, d_{c_3}) as the context, and the corresponding subgraph $\mathcal{G}_3$ from ConceptNet is retrieved (§ 3.2.1). 2) We insert the context node $\mathbf{z}_3^C$ into the graph and perform an attentive knowledge fusion (§ 3.2.2), forming a knowledge-enhanced graph $\tilde{G}_3$. 3) The graph view $\hat{G}_3$ for $\tilde{G}_3$ is generated through adaptive augmentation (§ 3.3.1). 4) We perform edge-scattered reasoning over graphs obtaining graph representations $\mathbf{z}_3^{\tilde{G}}, \mathbf{z}_3^{\hat{G}}$ (§ 3.3.2). 5) We predict the answer (§ 3.4) and compute the combination loss in a mini-batch (§ 3.5).

ignored the inherent noise existing in the KGs. In contrast, our work mainly focuses on incorporating contextual descriptions to address the noise problem in the KGs for efficient reasoning.

Graph Contrastive Learning. Graph contrastive learning is an extension of contrastive learning (CL) on graph-structured data (You et al., 2020; Zhu et al., 2021a). The basic idea of CL is to pull semantically similar samples close and keep dissimilar samples apart (Hadsell et al., 2006). Recently, CL has become an emerging topic in self-supervised representation learning, e.g., visual representations in CV (Chopra et al., 2005; Zhuang et al., 2019; Tian et al., 2020; Chen et al., 2020; Henaff, 2020; Caron et al., 2020; He et al., 2020; Misra and Maaten, 2020) and sentence representations in NLP (Mnih and Kavukcuoglu, 2013; Gao et al., 2021; Zhang et al., 2021; Yan et al., 2021; Meng et al., 2021). Inspired by CL, some works (Velickovic et al., 2019; Peng et al., 2020; Hassani and Khasahmadi, 2020; Qiu et al., 2020; Zhu et al., 2021b; Yang et al., 2022) implemented graph-oriented applications. They constructed graph views through stochastic augmentations, and then learnt effective graph representations by contrasting positive graph pairs with negative counterparts. For instances, Velickovic et al. (2019) extended deep InfoMax (Hjelm et al., 2018) to graphs and achieved significant performance on node representations. To provide the graphs with more global information, Hassani and Khasahmadi (2020) performed graph augmentation via graph diffusion kernels. Yang et al. (2022) conducted GCL among views generated in different spaces including the hyperbolic space and the Euclidean space. However, all these methods concentrated on unsupervised graph representation learning. In contrast, our work leverages the GCL scheme into the CQA task to improve the graph representations and enhance the training signals.

3 Our Proposed KE-GCL

The KE-GCL model framework is shown in Figure 2. We elaborate on the details as follows.

3.1 Problem Formulation

Formally, the CQA task can be defined as follow. Given a question q and a candidate answer set C with M choices, i.e., $C = \{c_1, c_2, \ldots, c_M\}$, we need to choose, from the candidate set C, the best matching one for the given question q.

3.2 Knowledge Enhancement

3.2.1 Knowledge Representation

Context Encoder. Similar to Xu et al. (2021), we obtain the contextual descriptions of entities for the current QA pair (q, c_i). These two descriptions are denoted as d_q and d_{c_i}, respectively. Then, we utilize PLMs as the context encoder to extract hidden representations. Thus, the QA pair and its descriptions are mapped into contextual hidden representation $\mathbf{z}_i^C$ as follows,

$$\mathbf{z}_i^C = f_C \left(q \oplus c_i \oplus d_q \oplus d_{c_i} \right) \quad (1)$$

where f_C is the context encoder and $\oplus$ denotes concatenation operator.

Graph Embedding. From the ConceptNet knowledge source, we retrieve the knowledge graph $\mathcal{G}_i$ for the QA pair (q, c_i) based on Yasunaga et al. (2021), which is the subgraph related to the entities in q and c_i. We use the pretrained entity weights from Feng et al. (2020) to initialize node embeddings $V_i = \{v_{i,1}, v_{i,2}, ..., v_{i,n}\}$. For the edges, we concatenate one-hot vectors of two node types and their edge relational type to initialize the edge embeddings, i.e., $[u_s \oplus r_{st} \oplus u_t]$. Then we use a two-layer MLP to encode the edge triplets into edge embeddings, denoted as $E_i = \left\{e_{i,1}, e_{i,2}, \cdots, e_{i,m}\right\}$. Thus, the directed graph is given as $G_i = (V_i, E_i), i \in [1, M]$ with n nodes and m edges.

3.2.2 Graph-oriented Knowledge Fusion

We perform knowledge fusion through node insertion and attention. Specifically, the node embeddings after insertion are updated as $\tilde{V}_i = \{v_{i,0}, v_{i,1}, \ldots, v_{i,n}\}$. Here, the context node embedding $v_{i,0} = f_M(\mathbf{z}_i^C)$, where the mapping f_M is a two-layer MLP. Moreover, those nodes related to the QA pair (q, c_i) are linked to the inserted node $v_{i,0}$, and the number of edges increases to $\tilde{m}$. Thus the edge embeddings are updated as $\tilde{E}_i = \left\{e_{i,1}, e_{i,2}, \cdots, e_{i,\tilde{m}}\right\}$. Then, we use attention mechanism to improve the knowledge fusion. The contextual representation $\mathbf{z}_i^C$ is used as a query to attend all the nodes. For each node $v_{i,q} \in \tilde{V}_i$, the attentive representation $\tilde{v}_{i,q}$ is given as,

$$\tilde{v}_{i,q} = \text{softmax}\left(\frac{f_Q(\mathbf{z}_i^C) \cdot v_{i,q}^{\top}}{\sqrt{D_g}}\right) \cdot v_{i,q} \tag{2}$$

where the mapping f_Q is a MLP and D_g is the dimension of node embedding. Thus, we obtain the knowledge-enhanced graph $\tilde{G}_i = (\tilde{V}_i, \tilde{E}_i), i \in [1, M]$ with $n+1$ nodes and $\tilde{m}$ edges.

3.3 Graph Contrastive Learning

3.3.1 Adaptive Graph Augmentation

Based on the knowledge-enhanced graph $\tilde{G}_i$, we construct an augmented view $\hat{G}_i$ for GCL through node-feature masking and edge dropping. Firstly, we define the influence of each node $\tilde{v}_{i,q} \in \tilde{V}_i$ as,

$$\rho_{\tilde{v}_{i,q}} = f_T(\tilde{v}_{i,q}) + f_R(\tilde{v}_{i,q}, \mathbf{z}_i^C) \tag{3}$$

in which $f_T(\cdot)$ and $f_R(\cdot,\cdot)$ represent the topological connectivity and contextual relevance, respectively. The topological connectivity is calculated by PageRank algorithm (Brin and Page, 1998), which weighs those nodes with more in-degrees. The contextual relevance is measured by cosine similarity, i.e.,

$$f_R(\tilde{v}_{i,q}, \mathbf{z}_i^C) = \theta(\tilde{v}_{i,q}, \mathbf{z}_i^C) = \frac{\tilde{v}_{i,q}^{\top}\mathbf{z}_i^C}{\left\|\tilde{v}_{i,q}\right\| \cdot \left\|\mathbf{z}_i^C\right\|} \tag{4}$$

which captures the semantic relevance with the contextual representation $\mathbf{z}_i^C$. Here, $\theta(\cdot,\cdot)$ denotes the cosine similarity between two vectors.

Secondly, we assume those dimensions frequently appearing in influential nodes should be important. Thus, the importance weight of dimension d for any node in $\tilde{V}_i$ is calculated as,

$$\gamma_{i,d} = \log \sum_{\tilde{v} \in \tilde{V}_i} |\tilde{v}[d]| \cdot \rho_{\tilde{v}} \tag{5}$$

Then we normalize the weight $\gamma_{i,d}$ as the probability for whether to mask the node dimension.

For each edge e in $\tilde{E}_i$, its importance depends on the importance weight of tail node $\tilde{v}_t$ which the edge points to, denoted as $\eta_e = \log \rho_{\tilde{v}_t}$. Likewise, we normalize the weight η_e as the probability for whether to drop edge e. Thus, we obtain the augmented view $\hat{G}_i = (\hat{V}_i, \hat{E}_i)$ of $\tilde{G}_i$ through sampling with these normalized probabilities.

3.3.2 Graph Reasoning

Both the knowledge-enhanced graph $\tilde{G}_i$ and its augmented view $\hat{G}_i$ are performed the same reasoning in this section. Taking the former $\tilde{G}_i = \{\tilde{V}_i, \tilde{E}_i\}$ as an example, we reason over the graph via edge scattering and attention-based node aggregating.

Specifically, to utilize the edge information, for each node $\tilde{v}_t \in \tilde{V}_i$, we obtain its initial hidden representation $h_t^{(0)}$ by scattering those edges which point to the node $\tilde{v}_t$,

$$h_t^{(0)} = \sum_{s \in \mathcal{N}_t} e_{st} + \tilde{v}_t \tag{6}$$

where $\mathcal{N}_t$ represents the neighbors of $\tilde{v}_t$. Then we use GAT to propagate and aggregate messages between nodes. In each layer $\ell \in [1, L]$, we update the representation of $\tilde{v}_t$ as follows,

$$h_t^{(\ell+1)} = \|_{u=1}^{U} \text{ReLU}\left(\sum_{s \in \mathcal{N}_t \cup \{t\}} \alpha_{st}^u W^u h_s^{(\ell)}\right) \tag{7}$$

where U is the number of attention heads, W^u is the corresponding linear projection matrix, and $\|$ is the concatenation operator for multiple heads. In addition, α_{st}^{u} is the attentive weight that scales each message from $\tilde{v}_s$ to $\tilde{v}_t$, which is given as,

$$\alpha_{st}^{u} = \frac{\exp\left(\gamma_{st}^{u}\right)}{\sum_{s' \in \mathcal{N}_t \cup \{t\}} \exp\left(\gamma_{s't}^{u}\right)} \tag{8}$$

and $\gamma_{st}^{u} = \mathrm{LeakyReLU}\left(W_{\ell}^{\top}\left[h_s^{(\ell)}, h_t^{(\ell)}\right]\right)$ reflects the relevant importance between these two nodes, and W_ℓ is a linear projection matrix in the ℓ-th layer.

After L-layer graph reasoning, we choose the hidden states of the context node as the pooling of the entire knowledge graph, i.e.,

$$\mathbf{z}_i^{\tilde{G}} = \mathrm{Pool}\left(h_0^{(L)}, h_1^{(L)}, \cdots, h_n^{(L)}\right) = h_0^{(L)}. \tag{9}$$

With the above adaptive graph augmentation and graph reasoning, we are ready for constrastive learning, which will be illustrated in Section 3.5.

3.4 Answer Prediction

For the choice c_i, we calculate its probability of being the correct answer using contextual representation $\mathbf{z}_i^C$ and graph representation $\mathbf{z}_i^{\tilde{G}}$,

$$P\left(c_i \mid q\right) = g_i \odot \left[\mathbf{z}_i^C W^C, \mathbf{z}_i^{\tilde{G}} W^{\tilde{G}}\right], \tag{10}$$

in which, the symbol $\odot$ denotes the element-wise product; W^C and $W^{\tilde{G}}$ denote the linear projection matrices. In addition, the gate g_i is given as,

$$g_i = \mathrm{softmax}\left(\mathrm{MLP}([\mathbf{z}_i^C, \mathbf{z}_i^{\tilde{G}}])\right). \tag{11}$$

The gate g_i is to control the importance weight of the context and the graph. For answer prediction, we calculate the probabilities for all candidate choices, and choose the most plausible answer with maximum probability score, i.e., $\mathrm{argmax}_{c_i \in C}\, p(c_i \mid q)$.

3.5 Training Objective

We train our KE-GCL model in an end-to-end fashion by minimizing the total loss $\mathcal{L}_T$ as follows,

$$\mathcal{L}_T = \mathcal{L}_{CE} + \lambda \mathcal{L}_{CL} \tag{12}$$

where $\mathcal{L}_{CE}$ and $\mathcal{L}_{CL}$ denote the answer prediction loss and the graph contrastive loss, respectively. In addition, λ is a tunable hyper-parameter to control the importance weight of GCL objective.

For the answer prediction loss, a standard cross-entropy loss is utilized to maximize the probability of the correct answer c_i,

$$\mathcal{L}_{CE} = -\log \frac{\exp\left(P(c_i \mid q)\right)}{\sum_{c_{i'} \in C} \exp\left(P(c_{i'} \mid q)\right)}. \tag{13}$$

For the graph contrastive loss, we describe the details in the following section.

3.5.1 Graph Contrastive Loss

We devise the graph contrastive loss based on the InfoNCE (Van den Oord et al., 2018). Moreover, we incorporate hard negatives to improve GCL. Intuitively, for a given question, the knowledge graph of candidate choices and their augmented views usually share some nodes and edges. Thus, there exist certain similarities among these graphs.

To this end, we set our hard negatives from two sources. One is the knowledge-enhanced graphs of those QA pairs with incorrect answers; the other is their corresponding augmented views. Formally, for a question q and its correct answer c_i, the graph representation is given as $\mathbf{z}_i{}^{\tilde{G}}$ using Eq. (9). Similarly, the graph representation of its augmented view is obtained as $\mathbf{z}_i{}^{\hat{G}}$. Then, the positive set is defined as $\mathcal{P} = \{\mathbf{z}_i^{\tilde{G}}, \mathbf{z}_i^{\hat{G}}\}$. Furthermore, with above intuition, we give our hard negative set as $\mathcal{N}_H = \{\mathbf{z}_j^{\tilde{G}} : j \neq i\} \bigcup \{\mathbf{z}_j^{\hat{G}} : j \neq i\}$. In addition, all QA pairs and their augmented views except the question under consideration are taken as common negatives. The common negative set is defined as $\mathcal{N}_C = \{\zeta_k^{\tilde{G}}\} \bigcup \{\zeta_k^{\hat{G}}\}$, in which two versions of ζ_k denote the corresponding graph representations of k-th QA pair in the mini-batch. Here, $k \in [1, M(N-1)]$ and N denotes the mini-batch size. Then, our negative set is composed of two sets, the hard negative set and common negative set.

Consequently, we design our graph contrastive loss as follows,

$$\mathcal{L}_{CL} = -\log \frac{T_{\mathcal{P}}}{T_{\mathcal{P}} + \beta T_{\mathcal{N}_H} + T_{\mathcal{N}_C}} \tag{14}$$

in which the positives contribution term is given as

$$T_{\mathcal{P}} = \exp^{\theta\left(\mathbf{z}_i^{\tilde{G}}, \mathbf{z}_i^{\hat{G}}\right)/\tau}, \tag{15}$$

the hard negatives contribution term is defined as

$$T_{\mathcal{N}_H} = \sum_{j=1, j\neq i}^{N} \exp^{\theta\left(\mathbf{z}_i^{\tilde{G}}, \mathbf{z}_j^{\tilde{G}}\right)/\tau} + \sum_{j=1, j\neq i}^{N} \exp^{\theta\left(\mathbf{z}_i^{\tilde{G}}, \mathbf{z}_j^{\hat{G}}\right)/\tau}, \tag{16}$$

and the term for common negatives contribution is formulated as

$$T_{\mathcal{N}_C} = \sum_{k=1}^{N(M-1)} \exp^{\theta\left(\mathbf{z}_i^{\tilde{G}}, \zeta_k^{\tilde{G}}\right)/\tau} + \sum_{k=1}^{N(M-1)} \exp^{\theta\left(\mathbf{z}_i^{\tilde{G}}, \zeta_k^{\hat{G}}\right)/\tau}. \tag{17}$$

In addition, β is the weighting factor for the hard negatives and τ denotes the temperature factor.

Dataset	Train	Dev	Test	# Choices
CommonsenseQA	9,741	1,221	1,140	5
OpenBookQA	4,957	500	500	4

Table 1: Statistics of CommonsenseQA and OpenBookQA datasets used in our evaluation.

4 Experiments and Results

4.1 Datasets and Metric

We evaluate our model on two benchmark datasets, i.e., CommonsenseQA (Talmor et al., 2019) and OpenbookQA (Mihaylov et al., 2018). The **CommonsenseQA** dataset creates questions from ConceptNet entities and relations, and contains 12,102 questions. CommonsenseQA involves a 5-way multiple choice QA task that requires reasoning with commonsense knowledge. The official test set of CommonsenseQA is not publicly available, therefore we perform experiments on the in-house (IH) data split used in Kagnet[4] (Lin et al., 2019). The **OpenBookQA** is built based on elementary science knowledge from an open book of 1,326 science facts, and contains 5,957 questions. It is a 4-way multiple choice QA task. We use the official data split of OpenbookQA[5]. The statistics of these two datasets are collected in Table 1. To evaluate the performance of CQA models, we use the **Accuracy** (i.e., Acc) as the metric.

4.2 Baselines

We compare our KE-GCL with state-of-the-art baselines, which are briefly reviewed as follows. 1) **RoBERTa-Large (w/o KG)** (Liu et al., 2019b) is based on an optimized BERT. The vanilla version only feeds the QA pair as the input and uses hidden states of the special token [CLS] to predict answers. 2) **Relation Network (RN)**[6] (Santoro et al., 2017) adapts multi-relational graph encoding. 3) **RGCN** (Schlichtkrull et al., 2018) is developed to deal with the highly multi-relational data of realistic knowledge bases. 4) **GconAttn** (Wang et al., 2019) presents a combination of techniques and utilizes external knowledge to improve the performance. 5) **KagNet** (Lin et al., 2019) is based on GCNs and LSTMs with a hierarchical path-based attention mechanism. 6) **MHGRN** (Feng et al., 2020) adopts the multi-hop graph relation network to perform reasoning by unifying path-based methods and GNNs. 7) **QA-GNN** (Yasunaga et al., 2021) performs joint reasoning over the QA context and KG with a joint graph representation.

[4]https://github.com/INK-USC/KagNet

[5]https://github.com/allenai/OpenBookQA

[6]In the experimental settings of RN, mean pooling is for 1-hop and attentive pooling is for 2-hop.

4.3 Implementation Details

For a fair comparison, we apply the same backbones using the Huggingface implementations[7] (Wolf et al., 2020) of PLMs, i.e., RoBERTa-Large for CommonsenseQA, and RoBERTa-Large and AristoRoBERTa[8] (Clark et al., 2020) for OpenBookQA. The hop size of retrieved subgraphs is set to 2. For the context encoder, we set the context dimension to 1024 and the max sequence length to 128. For the graph reasoning module, we set the graph dimension D_g to 200 and the number of GAT layers to 3. To control the influence of constraints in graph contrastive learning, we set the hyper-parameters λ, β and τ to 0.1, 2 and 0.2, respectively. The learning rate is set to 10^{-5} for the context encoder and 10^{-3} for other model components, the dropout rate is set to 0.2. We use RAdam (Liu et al., 2019a) as the model optimizer. We train the model in 30 epochs with an early stopping strategy, which takes about 13 hours using two GPUs (GeForce RTX 2080Ti) for a complete training procedure. In addition, we apply the gradient accumulation strategy (with a mini-batch size of 2) to achieve an equivalent effect of a batch size of 128. The reported results are the average on five runs with different random seeds.

4.4 Main Results

The experimental results on the two datasets CommonsenseQA and OpenBookQA are reported in Table 2 and Table 3, respectively. On these two datasets, our KE-GCL model consistently outperforms the other baseline models[9]. We observe that almost all the knowledge-aware models achieve performance gains over vanilla PLMs, which confirms the effectiveness of incorporating external knowledge in CQA task. Compared with the previous best model QA-GNN, our KE-GCL model surpasses its test performance by an average accuracy of 1.08% on the CommonsenseQA, (0.83% and 0.64%) on OpenBookQA datasets, respectively.

[7]https://github.com/huggingface/transformers

[8]AristoRoBERTa provides textual science facts for each question in OpenbookQA.

[9]We have also conducted experiments on CommonsenseQA with other backbones like ALBERT (Lan et al., 2019) and XLNet (Yang et al., 2019), but the results underperformed by 0.85~1.5 accordingly.

Method	IHdev-Acc	IHtest-Acc
RoBERTa-Large† (w/o KG) (Liu et al., 2019b)	70.70(±0.32)	67.23(±0.48)
+ RN (1-hop) (Santoro et al., 2017)	74.57 (±0.91)	69.08 (±0.21)
+ RN (2-hop) (Santoro et al., 2017)	73.65 (±3.09)	69.59 (±3.80)
+ RGCN (Schlichtkrull et al., 2018)	72.69 (±0.19)	68.41 (±0.66)
+ GconAttn (Wang et al., 2019)	72.61 (±0.39)	68.59 (±0.96)
+ KagNet (Lin et al., 2019)	73.47 (±0.22)	69.01 (±0.76)
+ MHGRN (Feng et al., 2020)	74.45 (±0.10)	71.11 (±0.81)
+ QA-GNN (Yasunaga et al., 2021)	76.54 (±0.21)	73.41 (±0.92)
+ KE-GCL (Ours)	**77.89** (±0.37)	**74.49** (±0.31)

Table 2: Performance comparison on the in-house development (IHdev) and in-house test (IHtest) sets of CommonsenseQA. The symbol "†" means that our reproduced result using the released code on the dataset.

Method	RoBERTa-Large	AristoRoBERTa
Fine-tuned LMs (w/o KG)	64.80 (±2.37)	78.40 (±1.64)
+ RN (1-hop) (Santoro et al., 2017)	63.65 (±2.31)	73.15 (±1.63)
+ RN (2-hop) (Santoro et al., 2017)	65.20 (±1.18)	75.35 (±1.39)
+ RGCN (Schlichtkrull et al., 2018)	62.45 (±1.57)	74.60 (±2.53)
+ GconAttn (Wang et al., 2019)	64.75 (±1.48)	71.80 (±1.21)
+ MHGRN (Feng et al., 2020)	66.85 (±1.19)	80.60 (±0.00)
+ QA-GNN (Yasunaga et al., 2021)	67.80 (±2.75)	82.77 (±1.56)
+ KE-GCL (Ours)	**68.63** (±1.24)	**83.41** (±1.93)

Table 3: Performance comparison on the official test set of OpenBookQA.

This demonstrates the effectiveness of our model for enhancing KG with contextual descriptions. Another finding is that in the OpenBookQA dataset, when changing the PLM from RoBERTa-Large to AristoRoBERTa, the performance is significantly improved. In addition, its performance still outperforms other baselines. This indicates that our KE-GCL model can effectively integrate the additional science facts to make better predictions.

Module	Setting	Acc
Our KE-GCL	Full Modules	**77.89**
Knowledge Enhancement	w/o ConceptNet	74.48 (3.41↓)
	w/o Wiktionary	75.14 (2.75↓)
	w/o Either	70.03 (7.86↓)
Graph Augmentation	w/o TC	77.27 (0.62↓)
	w/o CR	77.02 (0.87↓)
	w/o Either	76.78 (1.11↓)
Graph Reasoning	w/o Edge Scatter	77.55 (0.34↓)
	w/o GAT	77.14 (0.75↓)
	w/o Either	75.96 (1.93↓)
Graph Contrastive Learning	w/o $\mathcal{L}_{CL}$	75.66 (2.23↓)
	w/o Hard Neg	77.12 (0.77↓)

Table 4: Ablation study of our KE-GCL model on the CommonsenseQA IHdev set.

4.5 Ablation Study

To further investigate the effectiveness of the individual components in our KE-GCL, we conduct extensive ablation studies on the CommonsenseQA IHdev set. The results are reported in Table 4.

Knowledge Enhancement. We perform ablation on knowledge enhancement. When KGs from ConceptNet are removed ("w/o ConceptNet"), the performance decreases by 3.41%. However, when the contextual descriptions from Wiktionary are further removed ("w/o Either"), the performance significantly drops by 7.86%. It demonstrates the crucial role of enhancing KGs with contextual descriptions. These descriptions empower KGs with contextual understanding, which are beneficial for better knowledge coverage and answer inference.

Graph Augmentation. We perform ablation on the adaptive sampling strategy in graph augmentation. "w/o TC" and "w/o CR" are short for removing topological connectivity and contextual relevance when computing the sampling weights of nodes and edges, respectively. We observe that contextual relevance contributes more than the other. A possible reason is that those nodes with high contextual relevance is more likely to be intrinsically associ-

QA Example	Model	Predicted Choice	Predicted Score
Q1: Where can you find a snake in tall grass? A. tree B. in a jar C. pet shops D. field E. tropical forest	RoBERTa-Large	B. in a jar (✗)	[0.18, **0.24**, 0.18, 0.21, 0.19]
	QA-GNN	A. tree (✗)	[**0.41**, 0.11, 0.09, 0.18, 0.21]
	KE-GCL	D. field (✓)	[0.13, 0.11, 0.17, **0.46**, 0.13]
Q2: Sally was afraid of danger and always double checked what? A. fight enemy B. secure C. being safe D. safety E. vicinity	RoBERTa-Large	C. being safe (✗)	[0.14, 0.01, **0.45**, 0.23, 0.17]
	QA-GNN	D. safety (✓)	[0.09, 0.12, 0.34, **0.44**, 0.00]
	KE-GCL	D. safety (✓)	[0.02, 0.02, 0.28, **0.67**, 0.01]
Q3: What kind of service is my body a part of when I'm no longer here? A. bodycam B. home C. coffin D. funeral E. graveyard	RoBERTa-Large	A. bodycam (✗)	[**0.83**, 0.02, 0.00, 0.08, 0.07]
	QA-GNN	D. funeral (✓)	[0.05, 0.21, 0.08, **0.39**, 0.27]
	KE-GCL	B. home (✗)	[0.12, **0.40**, 0.19, 0.23, 0.06]
Q4: Where could you go to between 1000 and 10000 restaurant? A. big city B. town C. small town D. Canada E. yellow pages	RoBERTa-Large	D. Canada (✗)	[0.00, 0.12, 0.00, **0.73**, 0.15]
	QA-GNN	B. town (✗)	[0.08, **0.51**, 0.33, 0.01, 0.07]
	KE-GCL	B. town (✗)	[0.21, **0.33**, 0.27, 0.19, 0.00]

Table 5: Case study of the predicted choices and scores from 3 different models. The correct choices are underlined.

ated with the correct answer. Additionally, in "w/o Either" setting, we randomly sample nodes and edges for the augmented graph, leading to a decline of 1.11%. It further showcases the effectiveness of our adaptive sampling strategy.

Graph Reasoning. We perform ablation on the graph reasoning module, and dissect it into edge scatter and GAT components, denoted as "w/o Edge Scatter" and "w/o GAT", respectively. In both "w/o GAT" and "w/o Edge Scatter" settings, there is a slight decrease in the performance. However, when the entire reasoning module is removed ("w/o Either"), the performance suffers a decline of 1.93%, which is more serious than simply pooling these two effects together. This proves that our graph reasoning module can learn proper graph representations to infer answers via efficiently aggregating valuable messages from both nodes and edges.

Graph Contrastive Learning. We perform ablation on GCL. "w/o $\mathcal{L}_{CL}$" means we remove the objective for contrastive learning from the total loss $\mathcal{L}_T$, and "w/o Hard Neg" denotes we stop weighing hard negative graph pairs (i.e., those with incorrect choices) from all negatives in the mini-batch. In "w/o $\mathcal{L}_{CL}$" setting, the performance drops heavily by 2.23%. It confirms the important role of GCL in our model, because it can differentiate correct answers from other distractors by contrasting positive graph pairs with negative counterparts. Moreover, we find a 0.77% drop in "w/o Hard Neg" setting, which shows that setting hard negatives can bring further improvements for GCL.

To sum up, each component in our KE-GCL contributes to the entire performance of CQA task.

4.6 Attention Visualization

To illustrate the effectiveness of our graph augmentation strategy, we visualize the attention weights of a case from CommonsenseQA dataset. Specifically,

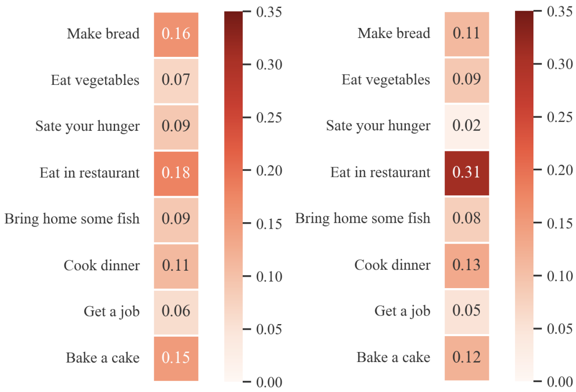

Figure 3: The attention heatmaps for the knowledge-enhanced graph and its augmented view are shown from left to right. The given question is "*Where is a human likely to go as a result of being hungry?*", and its correct answer is "*eat in restaurant*".

the attention weight is obtained from the context node to its 1-hop neighbors in the last layer of GAT. For the heatmaps in Figure 3, the QA pair is given as "*Where is a human likely to go as a result of being hungry?*" and "*eat in restaurant*". And the attention weights are shown on the sidebar for the knowledge-enhanced graph (left) and its augmented view (right). We observe that after augmentation, the context node gives more weights to "Eat in restaurant" which indicates the correct answer. While it pays less attention to other nodes such as "Sate your hunger" and "Make bread" which could be noisy for answer prediction. This demonstrates that our sampling strategy can effectively alleviate the noise in the graph. Thus, during graph reasoning, the model is inclined to focus on those favorable nodes for better answer prediction.

4.7 Case Study

We randomly select four cases from CommonsenseQA dataset in Table 5. The first two examples

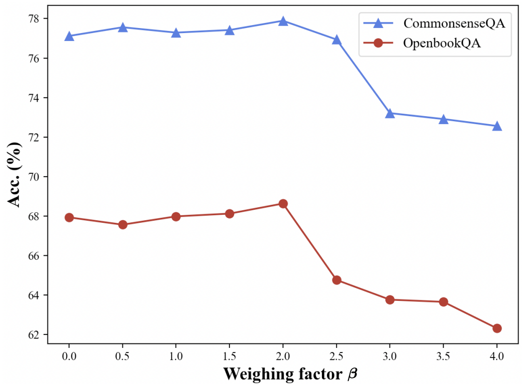

Figure 4: Effect of the factor β for hard negatives.

are of our correct prediction, and the other two examples are of our failed prediction. As for the correct cases, only our KE-GCL model makes the right prediction in the first example. The vanilla baseline without any external knowledge, i.e., RoBERTa-Large, produces a confusing result with a nearly uniform distribution. With the structural knowledge from ConceptNet, QA-GNN gives the strong but incorrect confidence to choice A ("tree"). The reason might be that the entity of "tree" is more closely related to the the entity of “tall grass” in the KG. In the second example, both QA-GNN and our KE-GCL model give the correct answer. However, in our solution, the gap between the correct choice D ("safety") and the disturbing choice C ("being safe") is further enlarged compared with QA-GNN. It indicates that the GCL strategy helps the model capture the nuances of similar choices.

As for the failure cases, only QA-GNN answers correctly in the third example. It seems that our KE-GCL model have not fully perceived the negation meanings contained in the question ("no longer"). Thus, the understanding deviation leads to a wrong prediction of choice B ("home"). In the last example, all these three models failed in prediction. The commonsense semantics lies in the hidden correlations between the number of restaurants and the corresponding choices. It reveals that there is still room for our KE-GCL model to understand numbers and handle numeric problems.

4.8 Effect of Hard Negatives in GCL

To investigate the impact of hard negatives in GCL, we evaluate our KE-GCL model with different values of weighing factor β in Eq. (14) on CommonsenseQA and OpenbookQA datasets. As shown in Figure 4, we can notice that when the weighing factor $\beta = 2.0$, the model achieves the best performance on both datasets. It demonstrates that setting appropriate hard negatives in GCL can indeed bring positive gains. On the one hand, when the value of β is small (i.e., < 2.0), the performance is relatively stable with a slight increase. On the other hand, when the value of weighing facto β climbs to more than 2.0, the performance sharply decreases. The model pays too much attention to the hard negatives. Therefore, the gradient for the positive graph pair is heavily weakened during back-propagation, resulting in the difficulty of optimizing the contrastive objective.

5 Conclusion and Future Work

In this paper, we propose a novel KE-GCL model for CQA task, which leverages contextual descriptions and GCL to reduce the noise in the KG. First, we integrate contextual descriptions into the KG, forming the knowledge-enhanced graph. Then, we devise an adaptive sampling strategy to generate the augmented view of the graph. Moreover, we reason over graphs via edge scattering and node aggregation. Finally, to further enhance the effect of GCL, we take graph pairs of incorrect answers as hard negatives. Extensive experiments on benchmark datasets verify that our KE-GCL model outperforms the baselines consistently. In the future, we will consider how to apply the GCL scheme in the few-shot or unsupervised scenarios for various CQA tasks.

Limitations

There remains at least one limitation in this study. Since our KE-GCL model is based on graph contrastive learning, a memory bank is needed for storing large volumes of negative graph pairs in the mini-batch. Therefore, using a larger mini-batch size to boost our KE-GCL's performance requires more GPU computation resources.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2019YFF0303302 and in part by the National Natural Science Foundation of China under Grant 62076032. We appreciate constructive feedback from the anonymous reviewers for improving the final version of this paper.

References

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2019. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations*.

Lisa Bauer, Yicheng Wang, and Mohit Bansal. 2018. Commonsense for generative multi-hop question answering tasks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4220–4230.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. 2020. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Sumit Chopra, Raia Hadsell, and Yann LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE.

Peter Clark, Oren Etzioni, Tushar Khot, Daniel Khashabi, Bhavana Mishra, Kyle Richardson, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, Niket Tandon, et al. 2020. From 'f'to 'a'on the ny regents science exams: An overview of the aristo project. *AI Magazine*, 41(4):39–53.

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.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Volume 2*, pages 1735–1742.

Kaveh Hassani and Amir Hosein Khasahmadi. 2020. Contrastive multi-view representation learning on graphs. In *International Conference on Machine Learning*, pages 4116–4126. PMLR.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Olivier Henaff. 2020. Data-efficient image recognition with contrastive predictive coding. In *International Conference on Machine Learning*, pages 4182–4192. PMLR.

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2018. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Yujia Li, Richard Zemel, Marc Brockschmidt, and Daniel Tarlow. 2016. Gated graph sequence neural networks. In *Proceedings of ICLR'16*.

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.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019a. On the variance of the adaptive learning rate and beyond. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *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*, volume 34, pages 8449–8456.

Yu Meng, Chenyan Xiong, Payal Bajaj, Paul Bennett, Jiawei Han, Xia Song, et al. 2021. Coco-lm: Correcting and contrasting text sequences for language model pretraining. *Advances in Neural Information Processing Systems*, 34.

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.

Todor Mihaylov and Anette Frank. 2018. Knowledgeable reader: Enhancing cloze-style reading comprehension with external commonsense knowledge. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 821–832.

Ishan Misra and Laurens van der Maaten. 2020. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717.

Andriy Mnih and Koray Kavukcuoglu. 2013. Learning word embeddings efficiently with noise-contrastive estimation. *Advances in neural information processing systems*, 26.

Zhen Peng, Wenbing Huang, Minnan Luo, Qinghua Zheng, Yu Rong, Tingyang Xu, and Junzhou Huang. 2020. Graph representation learning via graphical mutual information maximization. In *Proceedings of The Web Conference 2020*, pages 259–270.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1150–1160.

Lin Qiu, Yunxuan Xiao, Yanru Qu, Hao Zhou, Lei Li, Weinan Zhang, and Yong Yu. 2019. Dynamically fused graph network for multi-hop reasoning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6140–6150.

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. *Advances in neural information processing systems*, 30.

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*, pages 593–607. Springer.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

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.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer.

Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807.

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

Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep graph infomax. *ICLR (Poster)*, 2(3):4.

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

Dirk Weissenborn, Tomáš Kočiskỳ, and Chris Dyer. 2017. Dynamic integration of background knowledge in neural nlu systems. *arXiv preprint arXiv:1706.02596*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 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.

Wenhan Xiong, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. Improving question answering over incomplete kbs with knowledge-aware reader. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4258–4264.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. 2019. How powerful are graph neural networks? In *International Conference on Learning Representations*.

Yichong Xu, Chenguang Zhu, Ruochen Xu, Yang Liu, Michael Zeng, and Xuedong Huang. 2021. Fusing context into knowledge graph for commonsense question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1201–1207.

Yuanmeng Yan, Rumei Li, Sirui Wang, Fuzheng Zhang, Wei Wu, and Weiran Xu. 2021. Consert: A contrastive framework for self-supervised sentence representation transfer. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5065–5075.

Haoran Yang, Hongxu Chen, Shirui Pan, Lin Li, Philip S Yu, and Guandong Xu. 2022. Dual space graph contrastive learning. In *Proceedings of the ACM Web Conference 2022*, pages 1238–1247.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

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.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33:5812–5823.

Dejiao Zhang, Shang-Wen Li, Wei Xiao, Henghui Zhu, Ramesh Nallapati, Andrew O Arnold, and Bing Xiang. 2021. Pairwise supervised contrastive learning of sentence representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5786–5798.

Yanqiao Zhu, Yichen Xu, Qiang Liu, and Shu Wu. 2021a. An empirical study of graph contrastive learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021b. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, pages 2069–2080.

Chengxu Zhuang, Alex Lin Zhai, and Daniel Yamins. 2019. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6002–6012.