DECKER: Double Check with Heterogeneous Knowledge for Commonsense Fact Verification

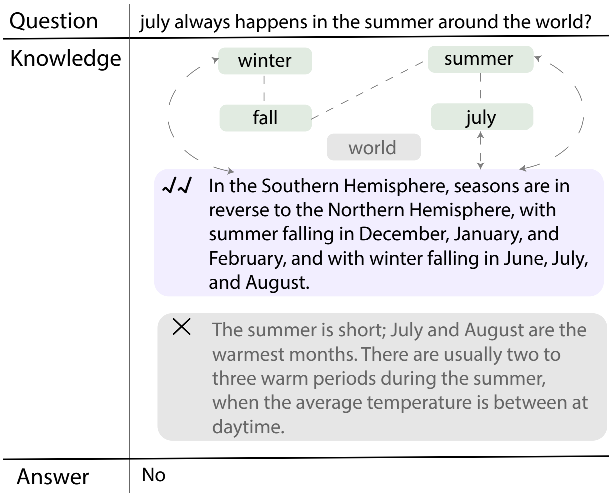
Anni Zou[1,2], **Zhuosheng Zhang**[1,2], **Hai Zhao**[1,2,*]
[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University
{annie0103,zhangzs}@sjtu.edu.cn,zhaohai@cs.sjtu.edu.cn

Abstract

Commonsense fact verification, as a challenging branch of commonsense question-answering (QA), aims to verify through facts whether a given commonsense claim is correct or not. Answering commonsense questions necessitates a combination of knowledge from various levels. However, existing studies primarily rest on grasping either unstructured evidence or potential reasoning paths from structured knowledge bases, yet failing to exploit the benefits of heterogeneous knowledge simultaneously. In light of this, we propose DECKER, a commonsense fact verification model that is capable of bridging heterogeneous knowledge by uncovering latent relationships between structured and unstructured knowledge. Experimental results on two commonsense fact verification benchmark datasets, CSQA2.0 and CREAK demonstrate the effectiveness of our DECKER and further analysis verifies its capability to seize more precious information through reasoning. The official implementation of DECKER is available at https://github.com/Anni-Zou/Decker.

1 Introduction

Commonsense question answering is an essential task in question answering (QA), which requires models to answer questions that entail rich world knowledge and everyday information. The major challenge of commonsense QA is that it not only requires rich background knowledge about how the world works, but also demands the ability to conduct effective reasoning over knowledge of various types and levels (Hudson and Manning, 2018). Recently, there emerges a challenging branch of commonsense QA: commonsense fact verification, which aims to verify through facts whether a given commonsense claim is correct or not (Onoe et al., 2021; Talmor et al., 2022). Different from previous *multiple-choice* settings which contain candidate answers (Talmor et al., 2019), commonsense fact verification solely derives from the question itself and implements reasoning on top of it (Figure 1). Therefore, it poses a novel issue of how to effectively seize the useful and valuable *knowledge* to deal with commonsense fact verification.



✓✓ In the Southern Hemisphere, seasons are in reverse to the Northern Hemisphere, with summer falling in December, January, and February, and with winter falling in June, July, and August.

✕ The summer is short; July and August are the warmest months. There are usually two to three warm periods during the summer, when the average temperature is between at daytime.

Figure 1: An example from CSQA2.0 (Talmor et al., 2022). Given the question, we perform a double check between the heterogeneous knowledge (i.e., KG and facts) and aim to derive the answer by seizing the valued information through reasoning.

One of the typical methods is to make direct use of knowledge implicitly encoded in pre-trained language models (PLMs) (Devlin et al., 2019; Liu et al., 2019; He et al., 2021), which have proved to be useable knowledge bases (Petroni et al., 2019; Bosselut et al., 2019). The knowledge in PLMs is gained during the pre-training stage through mining large-scale collection of unstructured text corpora. Nevertheless, the sore spot lies in that it is natural for human brains to project our prior world knowledge onto the answers facing the commonsense questions (Lin et al., 2019; Choi, 2022), whereas it is tough for PLMs to learn commonsense knowl-

[*] Corresponding author. This paper was partially supported by Key Projects of National Natural Science Foundation of China (U1836222 and 61733011).

Findings of the Association for Computational Linguistics: ACL 2023, pages 11891–11904
July 9-14, 2023 ©2023 Association for Computational Linguistics

edge that is implicitly stated in plain texts from corpora (Gunning, 2018).

To strengthen PLMs to perform commonsense QA, there is a surging trend of methods equipping language models with different levels of external knowledge, encompassing structured knowledge such as knowledge graphs (KG) (Lin et al., 2019; Yan et al., 2021; Yasunaga et al., 2021; Zhang et al., 2022b) and unstructured knowledge such as text corpus (Lin et al., 2021; Yu et al., 2022). While the KG-based methods yield remarkable performances on commonsense QA recently, they are more suitable and adaptive for *multiple-choice* settings because they lay emphasis on discovering connected patterns between the question and candidate answers. For example, to answer a question *crabs live in what sort of environment?* with candidate answers *saltwater*, *galapagos* and *fish market*, the KG-based methods manage to capture the path *crab–sea–saltwater* in KG, leading to a correct prediction. Nonetheless, they encounter a bottleneck when dealing with commonsense fact verification. Figure 1 shows an example: when asked whether *july always happens in the summer around the worlds*, the KG-based methods have a tendency to detect a strong link between *july* and *summer*, which may persuade the model to deliver the wrong prediction.

In general, there are two major limitations in previous studies. On one hand, structured knowledge abounds with structural information among the entities but suffers from sparsity and limited coverage. On the other hand, unstructured knowledge provides rich and broad context-aware information but undergoes noisy issues. These two kinds of knowledge can be naturally complementary to each other. However, most existing works focus on either structured or unstructured external knowledge but fail to exploit the benefits of heterogenous knowledge simultaneously. As the example in Figure 1 shows: if we rely only on the structured knowledge in KG, we tend to derive that *july* and *summer* are strongly correlated, with an extremely weak relationship between *summer* and *winter*. Similarly, if we focus only on the textual facts, we are more inclined to focus on the fact in grey, as it describes more information about *summer* in *july*. As a consequence, uncovering latent relationships among heterogeneous knowledge helps bridge the gap and yield more valuable and useful information.

Motivated by the above ideas, we propose DECKER, a commonsense fact verifier that bridges heterogeneous knowledge and performs a double check based on interactions between structured and unstructured knowledge. Our proposed DECKER works in the following steps: (i) firstly, it retrieves heterogeneous knowledge including a KG subgraph and several relevant facts following prior works (Zhang et al., 2022b; Izacard et al., 2022); (ii) secondly, it constructs an integral graph with encoded question and facts and then employs relational graph convolutional networks (R-GCN) to reason and filter over the heterogenous knowledge; (iii) lastly, it adopts a multi-head attention pooling mechanism to obtain a final refinement of enriched knowledge representation and combines it with the question representation for downstream tasks.

Our contributions are summarized as follows:

(i) For the concerned commonsense fact verification task, we initialize the research that simultaneously takes heterogeneous knowledge into account.

(ii) We propose a novel method in terms of R-GCN to construct an integral graph that executes a double check between structured and unstructured knowledge and better uncovers the latent relationships between them.

(iii) Experimental results on two commonsense fact verification benchmarks show the effectiveness of our approach, verifying the necessity and benefits of heterogeneous knowledge integration.

2 Related Work

2.1 Commonsense QA

Commonsense QA is a long-standing challenge in natural language processing as it calls for intuitive reasoning about real-world events and situations (Davis and Marcus, 2015). As a result, recent years have witnessed a plethora of research on developing commonsense QA tasks, including SWAG (Zellers et al., 2018), Cosmo QA (Huang et al., 2019), HellaSwag (Zellers et al., 2019), CSQA (Talmor et al., 2019), SocialIQa (Sap et al., 2019) and PIQA (Bisk et al., 2020). However, these tasks primarily attend to *multiple-choice* settings, so that there usually exist potential reasoning paths which explicitly connect the question with candidate answers. This may cause the models to be susceptible to shortcuts during reasoning (Zhang et al., 2022b). Therefore, a novel branch of commonsense QA: commonsense fact verification has emerged to further exploit the limits of reasoning models, such as CREAK (Onoe et al., 2021) and CSQA2.0 (Talmor et al., 2022). Unlike previous *multiple-choice* settings, commonsense fact verification needs the models to be granted richer background knowledge and higher reasoning abilities based on the question alone. Hence, our work dives into commonsense fact verification and conducts experiments on two typical benchmarks: CREAK and CSQA2.0.

2.2 Knowledge-enhanced Methods for Commonsense QA

Despite the impressive performance of PLMs on many commonsense QA tasks, they struggle to capture sufficient external world knowledge about concepts, relations and commonsense (Zhu et al., 2022). Therefore, it is of crucial importance to introduce external knowledge for commonsense QA. Currently, there are two major lines of research based on the property of knowledge: structured knowledge (i.e., knowledge graphs) and unstructured knowledge (i.e., text corpus).

The first research line strives to capitalize on distinct forms of knowledge graphs (KG), such as Freebase (Bollacker et al., 2008), Wikidata (Vrandečić and Krötzsch, 2014), ConceptNet (Speer et al., 2017), ASCENT (Nguyen et al., 2021) and ASER (Zhang et al., 2022a). Commonsense knowledge is thus explicitly delivered in a triplet form with relationships between entities. An initial thread of works endeavors to discover potential reasoning paths between the question and candidate answers under *multiple-choice* settings, which have shown remarkable advances in structured reasoning and question answering. For example, KagNet (Lin et al., 2019) utilizes a hierarchical path-based attention mechanism and graph convolutional networks to cope with relational reasoning. MHGRN (Feng et al., 2020) modifies from graph neural networks to make it adaptable for multi-hop reasoning while HGN (Yan et al., 2021) conducts edge generation and reweighting to find suitable paths more efficiently. JointLK (Sun et al., 2022) performs joint reasoning between LM and GNN and uses the dynamic KGs pruning mechanism to seek effective reasoning. Furthermore, other research optimizes by enhancing the interaction between raw texts of questions and KG to achieve better performance and robustness. QA-GNN (Yasunaga et al., 2021) designs a relevance scoring to make the interaction more effective, whereas GreaseLM (Zhang et al., 2022b) leverages multiple layers of modality interaction operations to achieve deeper interaction. Nevertheless, the scope of commonsense knowledge is infinite, far beyond a knowledge graph defined by a particular pattern.

The second research line attempts to make use of unstructured knowledge with either prompting methods (Lal et al., 2022; Qiao et al., 2023) or information retrieval techniques (Lewis et al., 2020a). Maieutic prompting (Jung et al., 2022) infers a tree of explanations through abductive and recursive prompting from generations of large language models (LLMs), which incurs high inference costs due to paywalls imposed by LLMs providers. DrFact (Lin et al., 2021) retrieves the related facts step by step through an iterative process of differentiable operations and further enhances the model with an external ranker. Talmor et al. (2020) employs regenerated data to train the model to reliably perform systematic reasoning. RACo (Yu et al., 2022) utilizes a *retriever-reader* architecture as the backbone and retrieves documents from a large-scale mixed commonsense corpus. Xu et al. (2021) extracts descriptions of related concepts as additional input to PLMs. However, these works mainly focus on homogeneous knowledge and reason on top of it, ignoring the need to fuse multiple forms of knowledge. Unlike previous works, our model is dedicated to intuitively modeling the relations between heterogeneous knowledge, bridging the gap between them, and filtering the more treasured knowledge by exploiting their complementary nature, in an inference-cost-free pattern.

Besides, there are some works taking heterogeneous knowledge into account to deal with commonsense reasoning. For instance, Lin et al. (2017) mines various types of knowledge (including event narrative knowledge, entity semantic knowledge and sentiment coherent knowledge) and encodes them as inference rules with costs to tackle commonsense machine comprehension. Nevertheless, this work is principally based on semantic or sentiment analysis at the sentence level, seeking knowledge enrichment at various levels of granularity. Our approach, however, is more concerned with extending external sources of knowledge and creating connections between heterogeneous knowledge from distinct sources so that they may mutually filter each other.

3 Methodology

This section presents the details of our proposed approach. Figure 2 gives an overview of its archi-

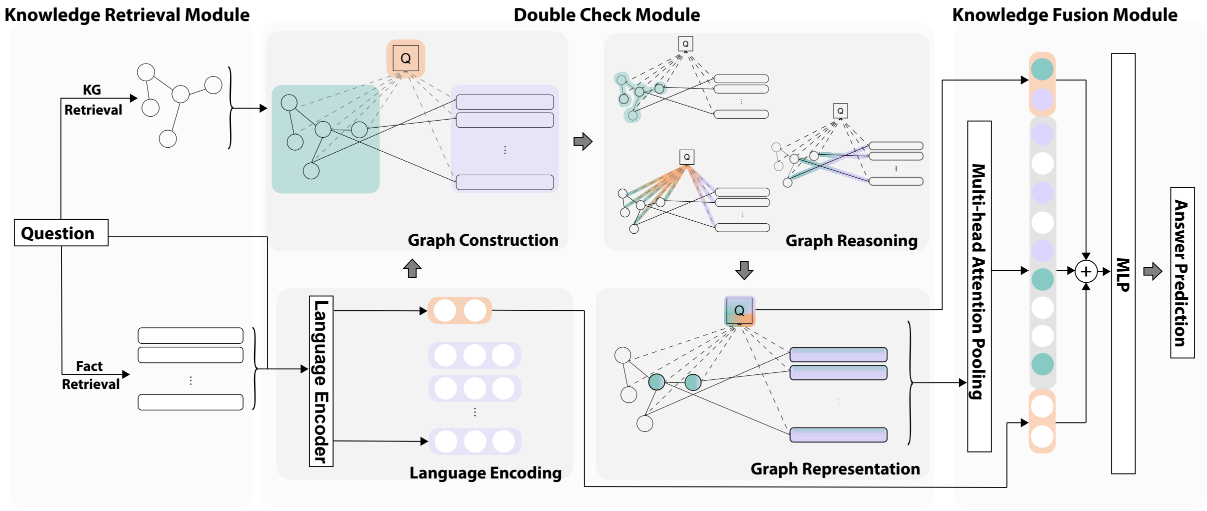


Figure 2: Overview of our approach, which consists of three components: Knowledge Retrieval Module (left), Double Check Module (middle), and Knowledge Fusion Module (right). Given an input question, KG retriever and fact retriever extract relevant local KG and facts (Knowledge Retrieval Module); then heterogeneous knowledge including entities in KG and facts are enhanced (Double Check Module); finally, heterogeneous knowledge is merged to deduce the final answer prediction (Knowledge Fusion Module).

tecture. Our approach, DECKER, consists of three major modules: (i) Knowledge Retrieval Module which retrieves heterogeneous knowledge based on the input question; (ii) Double Check Module which merges information from structured and unstructured knowledge and makes a double check between them; (iii) Knowledge Fusion Module which combines heterogeneous knowledge together to obtain a final representation.

3.1 Knowledge Retrieval Module

KG Retriever Given a knowledge graph $\mathcal{G}$ and an input question q, the goal of the KG Retriever is to retrieve a question-related sub-graph $\mathcal{G}_{sub}^{q}$ from $\mathcal{G}$. Following previous works (Lin et al., 2019; Yasunaga et al., 2021; Zhang et al., 2022b), we first execute entity linking to $\mathcal{G}$ to extract an initial set of nodes $\mathcal{V}_{init}$. We then obtain the set of retrieved entities $\mathcal{V}_{sub}$ by adding any bridge entities that are in a 2-hop path between any two linked entities in $\mathcal{V}_{init}$. Eventually, the retrieved subgraph $\mathcal{G}_{sub}$ is formed by retrieving all the edges that join any two nodes in $\mathcal{V}_{sub}$.

Fact Retriever Given a large corpus of texts containing K facts and an input question q, the objective of the fact retriever is to retrieve the top-k facts relevant to q. Following Contriever (Izacard et al., 2022) which is an information retrieval model pre-trained using the MoCo contrastive loss (He et al., 2020) and unsupervised data only, we employ a dual-encoder architecture where the question and facts are encoded independently by a BERT base uncased model (Huang et al., 2013; Karpukhin et al., 2020). For each question and fact, we apply average pooling over the outputs of the last layer to obtain its corresponding representation. Then a relevance score between a question and a fact is obtained by computing the dot product between their corresponding representations.

More precisely, given a question q and a fact $f_i \in \{f_1, f_2, \ldots, f_K\}$, we encode each of them independently using the same model. The relevance score $r(q, f_i)$ between a question q and a fact f_i is the dot product of their resulting representations:

$$r(q, f_i) = \langle E_\theta(q), E_\theta(f_i) \rangle\,, \qquad (1)$$

where $\langle , \rangle$ denotes the dot product operation and E_θ denotes the model parameterized by θ.

After obtaining the corresponding relevance scores, we select k facts $\mathcal{F} = \{f_q^1, f_q^2, \ldots, f_q^k\}$, whose relevance scores $r(q, f)$ are top-k highest among all K facts for each question q.

3.2 Double Check Module

Language Encoding Given a question q and a set of retrieved facts $\mathcal{F} = \{f_q^1, f_q^2, \ldots, f_q^k\}$, we deliver their corresponding sets of tokens $\mathcal{Q} = \{q^1, q^2, \ldots, q^t\}$ and $f_q^i = \{t_i^1, t_i^2, \ldots, t_i^{o_i}\}$ into a PLM, where t and o_i are the lengths of the question and fact sequence f_q^i, respectively. We obtain their

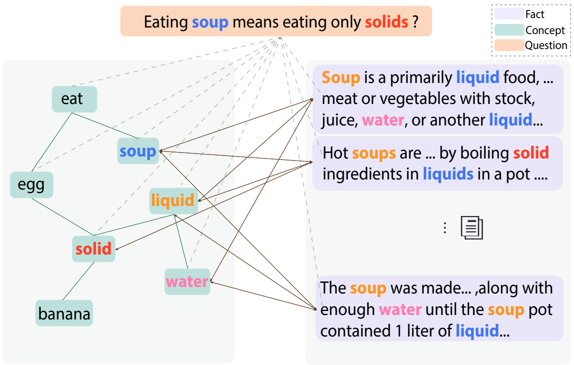

Eating soup means eating only solids ?
Fact
Concept
Question
eat
soup
egg
liquid
solid
water
banana
Soup is a primarily liquid food, ... meat or vegetables with stock, juice, water, or another liquid...
Hot soups are ... by boiling solid ingredients in liquids in a pot ....
The soup was made... ,along with enough water until the soup pot contained 1 liter of liquid...

Figure 3: An example of the constructed integral graph.

representations independently by extracting [CLS] inserted at the beginning:

$$\begin{aligned} q_{enc} &= \text{Encoder}\left(\{q^1, q^2, \ldots, q^t\}\right) \in \mathcal{R}^d, \\ f_{enc}^i &= \text{Encoder}\left(\{t_i^1, t_i^2, \ldots, t_i^{o_i}\}\right) \in \mathcal{R}^d, \\ \mathcal{F}_{enc} &= \left\{f_{enc}^1, f_{enc}^2, \ldots, f_{enc}^k\right\} \in \mathcal{R}^{k \times d}, \end{aligned} \tag{2}$$

where d denotes the hidden size defined by PLM.

Graph Construction Figure 3 gives an example of the constructed graph, which is dubbed as *integral graph*. Given a question q, a sub-graph $\mathcal{G}_{sub}^q$ extracted from KG and several retrieved facts $\mathcal{F} = \{f_q^1, f_q^2, \ldots, f_q^k\}$, we construct an integral graph denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{R})$. Here $\mathcal{V} = \mathcal{V}_q \cup \mathcal{V}_c \cup \mathcal{V}_f$ is the set of entity nodes, where $\mathcal{V}_q$, $\mathcal{V}_c$ and $\mathcal{V}_f$ denote the *question node* (orange in Figure 3), *concept nodes* (green in Figure 3) and *fact nodes* (purple in Figure 3), respectively; $\mathcal{E}$ is the set of edges that connect nodes in $\mathcal{V}$; $\mathcal{R}$ is a set of relations representing the type of edges in $\mathcal{E}$. In the integral graph, we define four types of edges[1]:

- concept-to-fact edges: (n_c, r_{c2f}, n_f);
- concept-to-concept edges: (n_c, r_{c2c}, n_c);
- question-to-fact edges: (n_q, r_{q2f}, n_f);
- question-to-concept edges: (n_q, r_{q2c}, n_c),

where $n_q \in \mathcal{V}_q$, $n_c \in \mathcal{V}_c$, $n_f \in \mathcal{V}_f$ and $\{r_{c2f}, r_{c2c}, r_{q2f}, r_{q2c}\} \subseteq \mathcal{R}$.

For question-to-concept and question-to-fact edges which are bidirectional, we connect the question node with all the other nodes in the integral graph with regard to enhancing the information flow between the question and its related heterogeneous knowledge. For concept-to-concept edges which are directional, we keep the structured knowledge extracted from KG and do not distinguish the multiple relations inside the sub-graph, as our approach mainly concentrates on effective reasoning over heterogeneous knowledge. For concept-to-fact edges, we use string matching and add a bidirectional edge (n_c, r_{c2f}, n_f) between $n_c \in \mathcal{V}_c$ and $n_f \in \mathcal{V}_f$ with $r_{c2f} \in \mathcal{R}$ if the concept n_c can be captured in the fact n_f. For instance, there should exist an edge between the concept *soup* and the fact *soup is primarily a liquid food*. In this way, the noisy and peripheral information is filtered whereas the relevant and precious knowledge is intensified.

Afterward, we initialize the node embeddings in the integral graph $\mathcal{G}$. For the concept nodes, we follow the method of prior work (Feng et al., 2020; Zhang et al., 2022b) and employ pre-trained KG embeddings for the matching nodes, which is introduced in Section 4.2. Then the pre-trained embeddings go through a linear transformation to align the dimension:

$$\begin{aligned} \mathcal{C}_{emb} &= \{c^1, c^2, \ldots, c^m\} \in \mathcal{R}^{m \times d_c}, \\ \mathcal{C}_{graph} &= \mathcal{C}_{emb} W_c + b_c \in \mathcal{R}^{m \times d}, \end{aligned} \tag{3}$$

where m denotes the number of concept nodes in the sub-graph, d_c denotes the hidden size of pre-trained KG embeddings, $W_c \in \mathcal{R}^{d_c \times d}$ and $b_c \in \mathcal{R}^d$ are trainable transformation matrices and bias vectors respectively.

For the question nodes and fact nodes, we inject the corresponding encoded results from PLM in Equation 2. Consequently, we obtain the initial node embeddings $\mathcal{N}^{(0)} \in \mathcal{R}^{(1+k+m) \times d}$ for the integral graph:

$$\mathcal{N}^{(0)} = \left[{q_{enc}}^{(0)}; {\mathcal{F}_{enc}}^{(0)}; {\mathcal{C}_{graph}}^{(0)}\right]. \tag{4}$$

Graph Reasoning As our integral graph $\mathcal{G}$ is a multi-relational graph where distinct edge types serve as varied information exchange between disparate knowledge, the message-passing process from a source node to a target node should be aware of its relationship, *i.e.,* relation type of the edge. For example, the concept-to-fact edges help to implement a double check and filtering between concepts and facts whereas the concept-to-concept edges assist in discovering the structured information. To this end, we adopt relational graph convolutional network (R-GCN) (Schlichtkrull et al., 2018) to perform reasoning on the integral graph.

[1] We ignore fact-to-fact edges due to the reason that if a fact-to-fact edge is added when the two facts link to the same concept node, a performance drop will be observed on the CREAK dev set (89.5% -> 87.3%).

In each layer of R-GCN, the current node representations $\mathcal{N}^{(l)}$ are fed into the layer to perform a round of information propagation between nodes in the graph and yield novel representations:

$$\mathcal{N}^{(l+1)} = \text{R-GCN}\left(\mathcal{N}^{(l)}\right). \tag{5}$$

More precisely, the R-GCN computes node representations $h_i^{(l+1)} \in \mathcal{N}^{(l+1)}$ for each node $n_i \in \mathcal{V}$ by accumulating and inducing features from neighbors via message passing:

$$h_i^{(l+1)} = \sigma\left(\sum_{r\in\mathcal{R}}\sum_{j\in N_i^r}\frac{1}{c_{i,r}}W_r^{(l)}h_j^{(l)} + W_0^{(l)}h_i^{(l)}\right), \tag{6}$$

where $\mathcal{R}$ is the set of relations, which corresponds to four edge types in our integral graph. N_i^r denotes the set of neighbors of node n_i, which are connected to n_i under relation r, and $c_{i,r}$ is a normalization constant. $W_r^{(l)}$ and $W_0^{(l)}$ are trainable parameter matrices of layer l. σ is an activated function, which in our implementation is GELU (Hendrycks and Gimpel, 2016).

Finally, we access the graph output through an L-layer R-GCN:

$$N^{(L)} = \left[{q_{enc}}^{(L)}; {\mathcal{F}_{enc}}^{(L)}; {\mathcal{C}_{graph}}^{(L)}\right]. \tag{7}$$

3.3 Knowledge Fusion Module

Multi-head Attention Pooling Since the acquired heterogeneous knowledge is leveraged to help answer the question, further interaction between the question and the knowledge is needed to refine the double-checked knowledge. Following the idea of Zhang et al. (2022b), we introduce a multi-head attention pooling mechanism (MHA) to ulteriorly gather the question-related information:

$$\begin{aligned}
\text{Attn}(Q,K,V) &= \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,\\
\text{head}_t &= \text{Attn}\left(H_qW_t^Q, H_kW_t^K, H_kW_t^V\right),\\
\text{MHA}(H_q,H_k) &= [\text{head}_1,\ldots,\text{head}_N]\,W^O,
\end{aligned} \tag{8}$$

where $W_t^Q \in \mathcal{R}^{d\times d_q}$, $W_t^K \in \mathcal{R}^{d\times d_k}$, $W_t^V \in \mathcal{R}^{d\times d_v}$, $W^O \in \mathcal{R}^{hd_v\times d}$ are trainable parameter matrices, h is the number of attention heads. d_q, d_k, d_v denote the hidden sizes of the query vector, key vector and value vector, respectively.

Specifically, we employ the initial question embedding from PLM as the query and feed it into MHA together with the graph-encoded representations of facts and concepts [2]. We thus derive the pooled knowledge representation:

$$K_a = \text{MHA}\left(q_{enc},\ \left[\mathcal{F}_{enc}^{(L)}; \mathcal{C}_{graph}^{(L)}\right]\right) \in \mathcal{R}^d. \tag{9}$$

Answer Prediction In the end, we concatenate the initial question embeddings q_{enc}, the pooled knowledge representation K_a and the enriched question representation $q_{enc}^{(L)}$ and deliver it into a predictor to get a final answer prediction:

$$l = \text{MLP}\left([q_{enc}; K_a; q_{enc}^{(L)}]\right) \in \mathcal{R}, \tag{10}$$

where the predictor is a two-layer MLP with a tanh activation of size $(3d, d, nlabel)$, $nlabel$ denotes the number of labels, which equals to 2 in our commonsense fact verification setting. The model is optimized using the cross entropy loss.

4 Experiments

4.1 Datasets

We conduct the experiments on two commonsense fact verification datasets: CommonsenseQA2.0 (Talmor et al., 2022) and CREAK (Onoe et al., 2021). The metric for evaluation is accuracy (acc).

CommonsenseQA2.0 is a commonsense reasoning dataset collected through gamification. It includes 14,343 assertions about everyday commonsense knowledge. We use the original *train / dev / test* splits from Talmor et al. (2022).

CREAK is a dataset for commonsense reasoning about entity knowledge. It is made up of 13,000 English assertions encompassing 2,700 entities that are either true or false, in addition to a small contrast set. Each assertion is generated by a crowdworker based on a Wikipedia entity, which can be named entities, common nouns and abstract concepts. We perform our experiments using the *train / dev / test / contrast* splits from Onoe et al. (2021).

4.2 Experimental Setup

Retrieval Corpus We leverage the English Wikipedia dump as the retrieval corpus. For preprocessing Wikipedia pages, we utilize the same method as described in Karpukhin et al. (2020); Lewis et al. (2020b). We divide each Wikipedia page into separate 100-word paragraphs, amounting to 21,015,324 facts in the end.

[2] We use the initial question embedding from PLM because it can capture the original information about the question. To verify this, the query in MHA is replaced with the post-RGCN representation and a slight performance drop is observed (89.5% -> 89.2%) on the CREAK dev set.

Model	#Total Params.	Single-task Training	CREAK Test	CREAK Contra	CSQA2.0 Test
Human (Onoe et al., 2021)			-	92.2	-
GreaseLM (Zhang et al., 2022b)	∼359M	✓	77.5	-	-
UNICORN (Lourie et al., 2021)	∼770M	✗	79.5	-	54.9
T5-3B (Raffel et al., 2022)	∼ 3B	✗	85.1	70.0	60.2
RACo (Yu et al., 2022)	≥ 3B	✗	**88.6**	74.4	61.8
DECKER (**Ours**)	∼449M	✓	88.4	**79.2**	**68.1**

Table 1: Experimental results on the CREAK and CSQA2.0 datasets. The evaluation metric is accuracy (acc).

Knowledge Graph We use *ConceptNet* (Speer et al., 2017), a general-domain knowledge graph, as our structured knowledge source G. It has 799,273 nodes and 2,487,810 edges in total. Node embeddings are initialized using the entity embeddings prepared by Feng et al. (2020), which consists of four steps: (1) it first converts knowledge triples in the KG into sentences using pre-defined templates for each relation; (2) it then feeds these sentences into PLM to compute embeddings for each sentence; (3) after that, it extracts all token representations of the entity's mention spans in these sentences; (4) it finally mean pools over these representations and projects this pooled representation.

Implementation Details Our model is implemented using Pytorch and based on the Transformers Library (Wolf et al., 2020). We fine-tune DeBERTa-V3-Large as the backbone pretrained language model for DECKER, and the hyperparameter setting generally follows DeBERTa (He et al., 2021). We set the layer number of the R-GCN as 3, with a dropout rate of 0.1 applied to each layer. The number of retrieved facts is set to 5 due to the trade-off for computation resources. The maximum input sequence length is 256. The initial learning rate is selected in {5e-6, 8e-6, 9e-6, 1e-5} with a warm-up rate of 0.1. The batch size is selected in {8, 16}. We run up to 20 epochs and select the model that achieves the best result on the development dataset.

4.3 Main Results

Table 1 presents the detailed results on two commonsense fact verification benchmarks: CREAK and CSQA 2.0. We compare our model with several baseline methods, which represent distinct knowledge-enhanced methods. UNICORN (Lourie et al., 2021) is instilled with external commonsense knowledge during the pre-training stage. GreaseLM (Zhang et al., 2022b) integrates structured knowledge into models during the fine-tuning stage. RACo (Yu et al., 2022) incorporates unstructured knowledge by constructing a commonsense corpus on which its retriever is trained [3]. Besides, we also compare our model with strong PLMs such as T5-3B (Raffel et al., 2022).

Model	Accuracy
DECKER	**89.5**
Knowledge Retrieval	
w/o facts	87.8(↓ 1.7)
w/o knowledge graph	87.9(↓ 1.6)
w/o both	86.1(↓ 3.4)
Graph Construction	
w/o question node	89.3(↓ 0.2)
w/o edge type	87.6(↓ 1.9)
w/o concept-to-fact edges	88.1(↓ 1.4)
w/o question-to-fact edges	88.8(↓ 0.7)
w/o concept-to-concept edges	88.3(↓ 1.2)
w/o question-to-concept edges	89.1(↓ 0.4)

Table 2: Ablation study of our model for components in Knowledge Retrieval and Graph Construction modules on the CREAK development set.

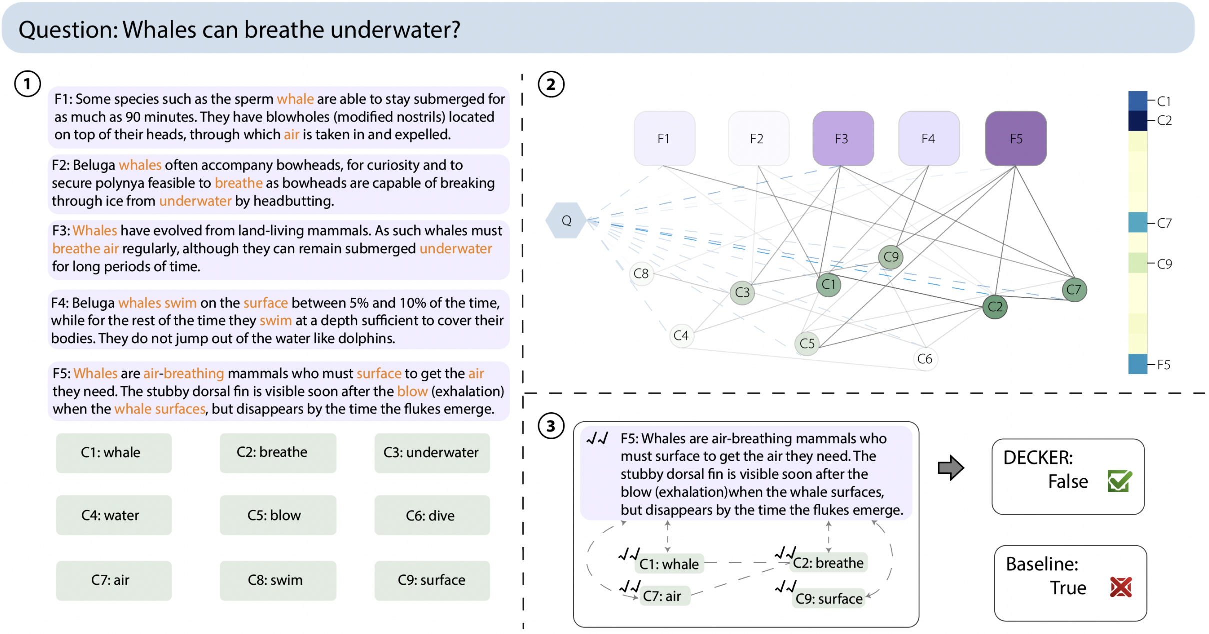
The results indicate that our model DECKER outperforms the strong baseline methods and achieves comparable results on the test set of CREAK. Besides, our model surpasses the current state-of-the-art model RACo on the contrast set of CREAK. Moreover, we observe that our model is lightweight and competitive without a considerable number of parameters and mixed data from multiple tasks during training, thus showing the strength and superiority of our model in various dimensions.

5 Analysis

5.1 Ablation Study

We conduct a series of ablation studies under the same set of hyperparameters to determine the contributions of key components in our model. Results

[3] RACo consists of two BERT-base models and T5-3B. The magnitude of the total parameter number depends largely on the latter, hence the sign of ≥ (greater than equal) is employed in Table 1.



Question: Whales can breathe underwater?

F1: Some species such as the sperm whale are able to stay submerged for as much as 90 minutes. They have blowholes (modified nostrils) located on top of their heads, through which air is taken in and expelled.

F2: Beluga whales often accompany bowheads, for curiosity and to secure polynya feasible to breathe as bowheads are capable of breaking through ice from underwater by headbutting.

F3: Whales have evolved from land-living mammals. As such whales must breathe air regularly, although they can remain submerged underwater for long periods of time.

F4: Beluga whales swim on the surface between 5% and 10% of the time, while for the rest of the time they swim at a depth sufficient to cover their bodies. They do not jump out of the water like dolphins.

F5: Whales are air-breathing mammals who must surface to get the air they need. The stubby dorsal fin is visible soon after the blow (exhalation) when the whale surfaces, but disappears by the time the flukes emerge.

F5: Whales are air-breathing mammals who must surface to get the air they need. The stubby dorsal fin is visible soon after the blow (exhalation)when the whale surfaces, but disappears by the time the flukes emerge.

DECKER: False ✓

Baseline: True ✗

Figure 4: An example showing how our model works to achieve the correct answer, in which our baseline fails. Texts in purple denote facts and texts in green denote concepts.

Model	CSQA2.0	CREAK
DeBERTa$_{large}$	67.9	86.1
DECKER	70.2(↑ 2.3)	89.5(↑ 3.4)

Table 3: Results on the CSQA2.0 and CREAK development sets. The evaluation metric is accuracy (acc).

Model	Interaction	Accuracy
DeBERTa$_{\text{LARGE}}$	✓	86.1
w/ max pooling	✗	87.5
w/ mean pooling	✗	86.7
w/ attention pooling	✓	88.9
w/ MHA pooling	✓	**89.5**

Table 4: Results of different pooling methods on the CREAK development set, MHA pooling denotes multi-head attention pooling for short.

in Table 2 demonstrate that the combination of heterogeneous knowledge and the components in our DECKER are both non-trivial. Results in Table 3 indicate that our DECKER outperforms the baseline by a large margin.

Knowledge Retrieval To investigate the effectiveness of knowledge combination, we discard the knowledge graph, facts and both. The resulting performances drop to 87.8%, 87.9%, and 86.1% respectively, which reveals the necessity of fusing knowledge with different granularity.

Graph Construction One of the crucial components of our model is graph construction, where the integral graph contains three types of nodes and four types of edges. We ablate the question node and remove all the edges connected with it. The results show that the removal hurts the performance. Furthermore, we dive into the edge analysis. We first treat all edges as the same type instead of four types, which witnesses a significant drop in performance. Our intuition is that effective reasoning among heterogenous knowledge should attend to edge types because they symbolize the distinct emphases during reasoning. We then erase each kind of edge respectively. Notably, the absence of concept-to-fact edges degrades the performance badly, suggesting the necessity of double-checking between heterogeneous knowledge.

5.2 Methods of Pooling

During the period of aggregating the graph output, we analyze the influence of different pooling methods, including max pooling, mean pooling, attention pooling and multi-head attention pooling. These pooling methods can be divided into two categories: those involving and those ignoring the interaction with the question. We compare the models with the same hyper-parameters on the development set of CREAK. Results in Table 4 demonstrate that the interaction process promotes the model performance, which may reveal that the graph reasoning executes more on the information flow between different levels of knowledge and the augmented inquiry about the initial question implements a final refinement of enriched knowledge. As shown in Table 4, employing multi-head attention pooling presents the best performance.

5.3 Interpretability: Case Study

In order to further explore the mechanism and get more intuitive explanations of our model, we select a case from CREAK in which the baseline model fails but our model succeeds. In addition, we analyze the node attention weights related to the question induced in MHA mechanism. Figure 4 shows that our DECKER can well bridge the reasoning between heterogeneous knowledge, thus leading to better filtering the noisy material and maintaining the beneficial information. Concretely, given the claim *whales can breathe underwater*, our model first extracts relevant structured and unstructured knowledge and then conducts reasoning over them. After reasoning, our model pays close attention to the concepts including *breathe*, *whale*, *air*, *surface* and the fact *whales are air-breathing mammals who must surface to get the air they need*, as shown in the attention heatmap. We can see that our model has the capability of manipulating heterogeneous knowledge to answer the questions.

6 Conclusion

In this work, we propose DECKER, a commonsense fact verification model that bridges heterogeneous knowledge and performs a double check based on the interactions between structured and unstructured knowledge. Our model not only uncovers latent relationships between heterogeneous knowledge but also conducts effective and fine-grained knowledge filtering of the knowledge. Experiments on two commonsense fact verification benchmarks (CSQA2.0 and CREAK) demonstrate the effectiveness of our approach. While most existing works focus on fusing one specific type of knowledge, we open up a novel perspective to bridge the gap between heterogeneous knowledge to gain more comprehensive and enriched knowledge in an intuitive and explicit way.

Limitations

There are three limitations. First, our model requires the retrieval of relevant structured and unstructured knowledge from different knowledge sources, which can be time-consuming. Using cosine similarity over question and fact embeddings can be a bottleneck for the model performance. Second, our model focuses on rich background knowledge but might ignore some inferential knowledge, which can be acquired from other sources such as Atomic. Third, our model might not be applicable to low resources languages where knowledge graphs are not available.

References

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

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

Yejin Choi. 2022. The Curious Case of Commonsense Intelligence. *Daedalus*, 151(2):139–155.

Ernest Davis and Gary Marcus. 2015. Commonsense reasoning and commonsense knowledge in artificial intelligence. *Commun. ACM*, 58(9):92–103.

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

Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. 2020. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online. Association for Computational Linguistics.

David Gunning. 2018. Machine common sense concept paper. *arXiv preprint arXiv:1810.07528*.

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.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual 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 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 2333–2338.

Drew A Hudson and Christopher D Manning. 2018. Compositional attention networks for machine reasoning. *arXiv preprint arXiv:1803.03067*.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*.

Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. 2022. Maieutic prompting: Logically consistent reasoning with recursive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1266–1279, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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

Yash Kumar Lal, Niket Tandon, Tanvi Aggarwal, Horace Liu, Nathanael Chambers, Raymond Mooney, and Niranjan Balasubramanian. 2022. Using commonsense knowledge to answer why-questions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1204–1219, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020a. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA. Curran Associates Inc.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. KagNet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2829–2839, Hong Kong, China. Association for Computational Linguistics.

Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William Cohen. 2021. Differentiable open-ended commonsense reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4611–4625, Online. Association for Computational Linguistics.

Hongyu Lin, Le Sun, and Xianpei Han. 2017. Reasoning with heterogeneous knowledge for commonsense machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2032–2043, Copenhagen, Denmark. Association for Computational Linguistics.

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

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13480–13488.

Tuan-Phong Nguyen, Simon Razniewski, and Gerhard Weikum. 2021. Advanced semantics for commonsense knowledge extraction. In *Proceedings of the Web Conference 2021*, pages 2636–2647.

Yasumasa Onoe, Michael JQ Zhang, Eunsol Choi, and Greg Durrett. 2021. Creak: A dataset for commonsense reasoning over entity knowledge. *arXiv preprint arXiv:2109.01653*.

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

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social IQa: Commonsense reasoning about social interactions. 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 4463–4473, Hong Kong, China. Association for Computational Linguistics.

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*.

Yueqing Sun, Qi Shi, Le Qi, and Yu Zhang. 2022. JointLK: Joint reasoning with language models and knowledge graphs for commonsense question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5049–5060, Seattle, United States. Association for Computational Linguistics.

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

Alon Talmor, Oyvind Tafjord, Peter Clark, Yoav Goldberg, and Jonathan Berant. 2020. Teaching pretrained models to systematically reason over implicit knowledge. *ArXiv*, abs/2006.06609.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2022. Commonsenseqa 2.0: Exposing the limits of ai through gamification. *arXiv preprint arXiv:2201.05320*.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A free collaborative knowledgebase. *Commun. ACM*, 57(10):78–85.

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

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, Online. Association for Computational Linguistics.

Jun Yan, Mrigank Raman, Aaron Chan, Tianyu Zhang, Ryan Rossi, Handong Zhao, Sungchul Kim, Nedim Lipka, and Xiang Ren. 2021. Learning contextualized knowledge structures for commonsense reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4038–4051, Online. Association for Computational Linguistics.

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

Wenhao Yu, Chenguang Zhu, Zhihan Zhang, Shuohang Wang, Zhuosheng Zhang, Yuwei Fang, and Meng Jiang. 2022. Retrieval augmentation for commonsense reasoning: A unified approach. *arXiv preprint arXiv:2210.12887*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of* *the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Hongming Zhang, Xin Liu, Haojie Pan, Haowen Ke, Jiefu Ou, Tianqing Fang, and Yangqiu Song. 2022a. Aser: Towards large-scale commonsense knowledge acquisition via higher-order selectional preference over eventualities. *Artificial Intelligence*, page 103740.

Xikun Zhang, Antoine Bosselut, Michihiro Yasunaga, Hongyu Ren, Percy Liang, Christopher D Manning, and Jure Leskovec. 2022b. Greaselm: Graph reasoning enhanced language models for question answering. *arXiv preprint arXiv:2201.08860*.

Chenguang Zhu, Yichong Xu, Xiang Ren, Bill Yuchen Lin, Meng Jiang, and Wenhao Yu. 2022. Knowledge-augmented methods for natural language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 12–20, Dublin, Ireland. Association for Computational Linguistics.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7 (Limitations)

☑ A2. Did you discuss any potential risks of your work?
7 (Limitations)

☑ A3. Do the abstract and introduction summarize the paper's main claims?
0, 1

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

B ☒ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
No response.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
No response.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
No response.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
No response.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
No response.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
No response.

C ☑ Did you run computational experiments?

Left blank.

□ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
No response.

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

□ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
No response.

□ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
No response.

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
No response.

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

Left blank.

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

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

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

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

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