Generated Knowledge Prompting for Commonsense Reasoning

Anonymous ACL submission

Abstract

It remains an open question whether incorporating external knowledge benefits commonsense reasoning while maintaining the flexibility of pretrained sequence models. To investigate this question, we develop generated knowledge prompting, which consists of generating knowledge from a language model, then providing the knowledge as additional input when answering a question. Our method does not require task-specific supervision for knowledge integration, or access to a structured knowledge base, yet it improves performance of large-scale, state-of-the-art models on four commonsense reasoning tasks, achieving state-of-the-art results on numerical commonsense (NumerSense), general commonsense (CommonsenseQA 2.0), and scientific commonsense (QASC) benchmarks. Generated knowledge prompting highlights large-scale language models as flexible sources of external knowledge for improving commonsense reasoning. Our code is available at `github.com/anonymous_repo`.

1 Introduction

It remains an open research question whether external knowledge is needed for commonsense reasoning. On one hand, a substantial body of prior work has reported that integrating external knowledge can help improve task performance (Mitra et al., 2019; Bian et al., 2021, *inter alia*), especially if the knowledge is high quality (e.g. hand-crafted by experts). On the other hand, recent leaderboards are often dominated by large-scale pretrained models that are fine-tuned on a target benchmark (Khashabi et al., 2020; Lourie et al., 2021), suggesting that the benefits of external knowledge may wash away as the underlying models increase in size and are pretrained on ever larger amounts of raw text.

Even if external knowledge is found to be effective on a particular task, *flexibility* remains a fundamental hurdle to integrating external knowledge, as many benchmarks currently lack appropriate knowledge bases with sufficient coverage. Furthermore, prior methods often require task-specific, custom supervision for knowledge integration (Mitra et al., 2019; Chang et al., 2020), introducing a burden for rapidly adapting new pretrained models to a wide variety of tasks.

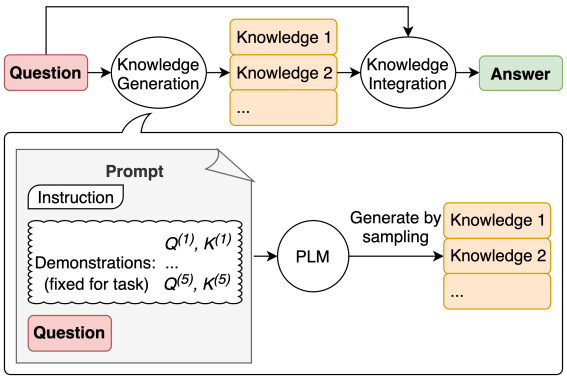

Figure 1: Generated knowledge prompting involves (i) using few-shot demonstrations to generate question-related knowledge statements from a language model; (ii) using a second language model to make predictions with each knowledge statement, then selecting the highest-confidence prediction.

In this paper, we investigate whether external knowledge can be helpful for commonsense reasoning, even on top of the largest state-of-the-art pretrained models (e.g. T5-11b (Raffel et al., 2019) and its variants), with a focus on four recent commonsense benchmarks. To facilitate easier adaptation with any zero-shot or finetuned models, we propose an approach that does not require access to a structured knowledge base or joint finetuning for knowledge integration.

The key insight behind our method, Generated Knowledge Prompting (sketched in Figure 1), is that we can generate useful knowledge from a language model, then provide the knowledge as an input prompt that is concatenated with a question. To

Dataset	Question / *Knowledge*	Prediction	Score
NumerSense	**the word children means [M] or more kids.** *The word child means one kid.*	one two	0.37 | 0.35 0.91
CSQA	**She was always helping at the senior center, it brought her what?** *People who help others are usually happier.*	feel better happiness	0.97 | 0.02 0.98
CSQA2	**Part of golf is trying to get a higher point total than others.** *The player with the lowest score wins.*	yes no	1.00 | 0.00 1.00
QASC	**Sponges eat primarily** *Sponges eat bacteria and other tiny organisms.*	cartilage krill and plankton	0.95 | 0.00 0.99

Table 1: Examples where prompting with generated knowledge rectifies model prediction. Each section shows the correct answer in green, the incorrect answer in red, and the prediction scores from the inference model that only sees the question (top) and the same model that sees the question prompted with the given knowledge (bottom).

support a variety of settings without finetuning, the quality and flexibility of knowledge is crucial. We propose a simple, yet effective, method that elicits *knowledge statements* (i.e. knowledge expressed as natural language statements) from generic language models in a few-shot setting. Compared to prior work that elicits knowledge via clarification questions (Shwartz et al., 2020) or contrastive explanations (Paranjape et al., 2021), our approach can generate knowledge flexibly, beyond the scope of pre-defined templates (Table 1).

Experiments show that our method improves both zero-shot and finetuned models on numerical commonsense (NumerSense (Lin et al., 2020)), general commonsense (CommonsenseQA (Talmor et al., 2019), CommonsenseQA 2.0 (Talmor et al., 2021)), and scientific commonsense (QASC (Khot et al., 2020)) benchmarks, setting a new state-of-the-art on three of these datasets. It outperforms the template-based knowledge generation method *self-talk* (Shwartz et al., 2020), while performing comparably to retrieval-based systems.

We find three factors contribute to the performance of generated knowledge prompting: (i) the *quality* of knowledge, (ii) the *quantity* of knowledge where the performance improves with more knowledge statements, and (iii) the strategy for integrating knowledge during inference. Our qualitative analysis suggests that the generated knowledge statements cover a variety of types, and can transform commonsense question answering to explicit reasoning procedures, e.g. deduction, that are supported by off-the-shelf and finetuned language models.

2 Generated Knowledge Prompting

A multiple-choice commonsense reasoning task involves predicting an answer $a \in A_q$ given a question $q \in Q$, where the set of choices A_q is finite and can vary by question, and both questions and answers are variable-length text sequences. Our method answers commonsense questions in two steps.

The first step is *knowledge generation*, where we use a language model $p_G(k|q)$ to generate knowledge statements conditioned on the question:

$$K_q = \{k_m : k_m \sim p_G(k|q), m = 1 \dots M\},$$

where each knowledge statement k_m is a variable-length text sequence. Intuitively, each statement contains information that is helpful for answering the question (e.g. Table 1).

The second step is *knowledge integration*, where generated knowledge is integrated into the decision process of a language model used for inference,

$$\hat{a} = \arg\max_{a \in A_q} p_I(a|q, K_q)$$

In contrast, the *vanilla* setting of using the inference model without knowledge is represented by $\hat{a} = \arg\max_{a \in A_q} p_I(a|q)$.

Next, we describe the knowledge generation and integration steps in detail.

2.1 Knowledge Generation

We generate question-related knowledge statements by prompting a language model. The prompt consists of an instruction, a few demonstrations that are fixed for each task, and a new-question placeholder. The demonstrations are human-written, and each consists of a question in the style of the task and a knowledge statement that is helpful for answering this question. For a given task, we write five demonstrations using the format in Table 2.

We write questions (or select them from the training set, when available) that are representative of

Task	NumerSense	QASC
Prompt	Generate some numerical facts about objects. Examples:	Generate some knowledge about the input. Examples:
	Input: **penguins have <mask> wings.** Knowledge: *Birds have two wings. Penguin is a kind of bird.*	Input: **What type of water formation is formed by clouds?** Knowledge: *Clouds are made of water vapor.*
	...	...
	Input: **a typical human being has <mask> limbs.** Knowledge: *Human has two arms and two legs.*	Input: **The process by which genes are passed is** Knowledge: *Genes are passed from parent to offspring.*
	Input: **{question}** Knowledge:	Input: **{question}** Knowledge:

Table 2: Prompts for knowledge generation for two of our tasks, NumerSense and QASC. The prompt consists of an instruction, five demonstrations of question-knowledge pairs, and a new question placeholder. For full prompts on all the tasks we evaluate on, see Appendix A.2.

challenges posed by the task (e.g. numerical commonsense, scientific commonsense). We pair each question with a knowledge statement that turns the commonsense problem posed by the question into an explicit reasoning procedure, without directly answering the question. For example, the knowledge statement *Birds have two wings. Penguin is a kind of bird.* is helpful for the question **Penguins have <mask> wings**, because it turns the problem into deductive reasoning. Meanwhile, *Penguins have two wings.* would be a poor knowledge statement to demonstrate according to our guideline.

When generating knowledge for a new question q, we plug the question into the placeholder, and repeatedly sample generated continuations of this prompt to obtain a set of knowledge statements $K_q = \{k_1, k_2, \ldots, k_M\}$. For full prompts on all the tasks we evaluate on, see Appendix A.2.

2.2 Knowledge Integration via Prompting

In the knowledge integration step, we use a language model – called the inference model – to make predictions with each generated knowledge statement, then select the highest-confidence prediction. Specifically, we use each knowledge statement to prompt the model, forming M knowledge-augmented questions:

$$q_0 = q, q_1 = [k_1||q], \ldots, q_M = [k_M||q]$$

where $[\cdot||\cdot]$ denotes text concatenation.

We compute an aggregated score for each answer choice a using the augmented question that best supports it under the inference model:

$$p_I(a|q, K_q) \propto \max_{0 \le m \le M} p_I(a|q_m). \quad (1)$$

Intuitively, this favors knowledge statements that strongly support one of the choices.

The predicted answer is then,

$$\hat{a} = \arg\max_{a \in A_q} \max_{0 \le m \le M} p_I(a|q_m),$$

which is the choice that gets most support from one of the knowledge statements. This prediction uses a single knowledge statement, which we refer to as the *selected knowledge*:

$$\hat{k} = k_{\hat{m}} \text{ where } \hat{m} = \arg\max_{0 \le m \le M} \max_{a \in A_q} p_I(a|q_m).$$

The inference model may be any existing language model taken off-the-shelf (i.e. zero-shot) or finetuned on the task. We do not do any further finetuning with knowledge prompting.

3 Experimental Setup

Here, we describe the implementation details of our method and how they are adapted to each task.

For knowledge generation, we use GPT-3 (Brown et al., 2020) as the underlying language model, where our few-shot prompting method is most effective. We generate $M = 20$ knowledge statements for each question with nucleus sampling $p = 0.5$ (Holtzman et al., 2019), and discard repetitions and empty strings. Generation is terminated when it exceeds 64 tokens or hits the `\n` token.[1]

For inference, we use off-the-shelf T5 (Raffel et al., 2019) and GPT-3, as well as finetuned models that are state-of-the-art on each dataset, including UnifiedQA (UQA) (Khashabi et al., 2020) and Unicorn (Lourie et al., 2021). See details in the task setup below.

3.1 Datasets and Task Setup

We evaluate our method on four commonsense reasoning datasets which cover a variety of challenges and problem formats.

[1] An exception is with the CSQA2 dataset, where for the best results we choose $M = 5$ and allow for up to 128 tokens in each generation.

NumerSense (Lin et al., 2020) consists of numerical statements about common objects and concepts where for each sentence we need to recover a masked number word. The choices are integers ranging from zero to ten, plus the word *no*, so the task can be framed as a multiple-choice problem. Since NumerSense is a diagnostic dataset, we only use zero-shot inference models, which is the current SOTA. We follow Zhang (2021) who uses the state-of-the-art zero-shot T5 with text-infilling setup and select the choice with highest likelihood on its token(s). We also implement zero-shot GPT-3 inference, where we plug in each choice to the question and compute the choice probability as the generative probability of the entire sentence, normalized over all the choices.

CommonsenseQA (CSQA) (Talmor et al., 2019) is a 5-way multiple-choice QA dataset about common world scenarios. We do inference with the zero-shot and finetuned T5 models. For zero-shot T5, we format the question as text-infilling, and predict the choice with highest sequence-to-sequence language modeling probability. For finetuned T5 (including UnifiedQA which is SOTA), we use the same setup as Khashabi et al. (2020).

CommonsenseQA 2.0 (CSQA2) (Talmor et al., 2021) is a binary classification dataset where we need to judge whether commonsense statements are true or false. We only do inference with the finetuned model, due to poor calibration of zero-shot models on this dataset. We use finetuned Unicorn (Lourie et al., 2021), which is the current SOTA, following the setup in Talmor et al. (2021).

QASC (Khot et al., 2020) is an 8-way multiple-choice QA dataset about grade school science. This dataset also includes two pieces of background knowledge per question, whose composition fully answers the question. We do inference with zero-shot T5 and finetuned T5 (including UnifiedQA which is SOTA), using the same setups as CSQA.

3.2 Knowledge Generation Baselines

We study the impact of our knowledge generation method (shorthanded as K) by comparing with the following baselines:

No knowledge ($\varnothing$) We refer to inference without any knowledge statements as the *vanilla* baseline.

Random sentences (R) Sampling random sentences from the language model without conditioning on the question. We use the same implementation setup as our knowledge generation method (i.e. also using GPT-3, with the same hyperparameters).

Context sentences (C) Sampling sentences from the context of the question. This is implemented by sampling text continuations of the question from the language model. We use the same implementation setup as our knowledge generation method.

Template-generated knowledge (T) Self-talk (Shwartz et al., 2020) uses manually-designed templates to elicit knowledge statements from language models. For fair comparison, we use GPT-3 as the knowledge generator in self-talk, and bound the number of generations to $M = 20$ per question. Templates and other hyperparameters are kept the same as their original paper.

Retrieval-based knowledge (IR) Instead of being generated, knowledge can be retrieved from appropriate sources. We consider the following retrieval-based methods. For NumerSense, knowledge is retrieved from sentences in Wikipedia and GenericsKB. For CSQA2, we use snippets returned by Google when querying the question. For QASC, we use the associated fact sentences that are used to create each question.

4 Experimental Results

As we will show, our generated knowledge prompting method sets new state-of-the-art results on most datasets we evaluate on, and works well under both zero-shot and finetuned settings. In particular, our knowledge generation outperforms naive baselines as well as template-based knowledge generation, and is on-par with retrieval-based systems.

4.1 Overall Performance

Table 3 shows the results on zero-shot and finetuned models following our task setups.

New state-of-the-art. We apply our method on top of the same inference model used in the previous state-of-the-art. On NumerSense, we achieve a 6% (66.18 → 72.47) improvement over the previous best method based on the zero-shot T5 model. The previous state-of-the-art among non-retrieval methods on CSQA2 is based on the finetuned Unicorn model, upon which we improve by 2% (70.2 → 73.03). For QASC, the previous best is based on the finetuned UnifiedQA model, upon which we improve by 3% (76.74 → 80.33).

Zero-shot settings. Columns A, B_1, and D_1 in Table 3 show that our method substantially improves zero-shot inference models, by 7% to

		A			B_1	B_2	C		D_1		D_2	
	Dataset	**NumerSense**			**CSQA**	**CSQA**	**CSQA2**		**QASC**		**QASC**	
	Inference Model	T5-11b			T5-11b	UQA-11b-ft	Unicorn-ft		T5-11b		UQA-11b-ft	
		dev	$\text{test}_{\text{core}}$	test_{all}	dev	dev	dev	test	dev	test	dev	test
Knowledge Gen.	($\varnothing$) Vanilla baseline	67.5	70.23	64.05	39.89	85.18	69.9	70.2†	48.16	44.89	81.75	76.74
	(R) Random sentences	68.5	–	–	21.79	**85.42**	70.37	–	49.35	–	82.18	–
	(C) Context sentences	70.5	–	–	42.51	85.34	70.92	–	55.83	–	82.61	–
	(T) Template-based	–	–	–	45.37	–	–	–	–	–	–	–
	(IR) Retrieval-based	–	70.41	65.10**	–	–	**74.0**	**73.3**††	**76.89**	–	**90.06**	–
	(K) Ours	**78.0**	**79.24**	**72.47**	**47.26**	85.34	72.37	73.03	58.32	55.00	84.02	80.33
	prev. SOTA (no IR)	–	72.61	66.18*	–	79.1 (test)#	69.9	70.2†	–	–	81.75	76.74‡

Table 3: Experimental results of applying different knowledge generation methods on various tasks and inference models. T5-11b is the zero-shot inference model, whereas other inference models are finetuned based on T5-11b. We **bold** the best and underline the second best numbers. Previous SOTA and retrieval-based methods are also based on the inference model in their corresponding column: * T5-11b 1.1 +digits (Submission by ISI Waltham); ** T5-11b + IR (Yan, 2021); # UQA-11b-ft (Khashabi et al., 2020) (SOTA of single-model methods without referencing ConceptNet); † Unicorn-ft (Talmor et al., 2021); †† Unicorn-ft + Google snippets (Talmor et al., 2021); ‡ UQA-11b-ft (Khashabi et al., 2020).

10% across NumerSense (64.05 → 72.47), CSQA (39.89 → 47.26), and QASC (44.89 → 55.00).

Finetuned settings. Columns B_2, C, and D_2 in Table 3 indicate that our method consistently improves upon the vanilla baseline set by finetuned inference models (though by smaller margins than in the zero-shot settings).

4.2 Knowledge Generation Methods

Table 3 reports the performance with different knowledge generation baselines. Generally, random sentences barely help and even hurt the inference model, whereas context sentences of the question provide some gain. In contrast, knowledge generated by our method consistently leads to substantial performance improvements, which implies that our knowledge is of high quality.

Our knowledge outperform template generated knowledge. We compare our knowledge generation method with the template-based *self-talk* on the CSQA dev set. (CSQA is the only task we experiment with that has self-talk templates available.) Our method leads to a larger improvement over the T5-11b baseline than self-talk (by 1.89%), showing that it is better at eliciting helpful knowledge from models.

Our knowledge is comparable with retrieval-based knowledge. On NumerSense, the retrieved knowledge only improves inference performance by 0.18% on test-core and 1.02% on test-all, while our method further outperforms it by 8.83% and 7.37%, respectively. This shows

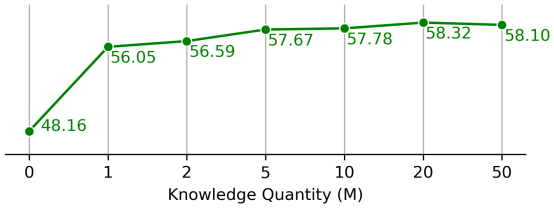

Figure 2: Performance with different number of generated knowledge statements per question. Measured on QASC dev set with the T5-11b inference model.

that knowledge retrieved from a loosely-related knowledge base can be far less useful than our generated knowledge. On CSQA2, although we are not able to beat the web-retrieved knowledge, our method still bridges the performance gap without referring to Google search. For QASC, the "retrieved" knowledge is actually gold knowledge from a knowledge base that was used to construct the dataset. As a result, our generated knowledge falls significantly short of the retrieved knowledge. In summary, our generated knowledge is roughly comparable with retrieved knowledge in terms of downstream performance, and is most valuable when there is no appropriate in-domain knowledge base to retrieve from.

4.3 Analysis

Better performance with more knowledge. We analyze the impact of the number of generated knowledge statements, M, and show the results in Figure 2. Generally, the performance increases

Integration method	QASC-dev
ours	**58.32**
Mixture-of-Experts	56.26
Product-of-Experts	55.94

Table 4: Performance with different knowledge integration methods (QASC dev set, T5-11b inference model).

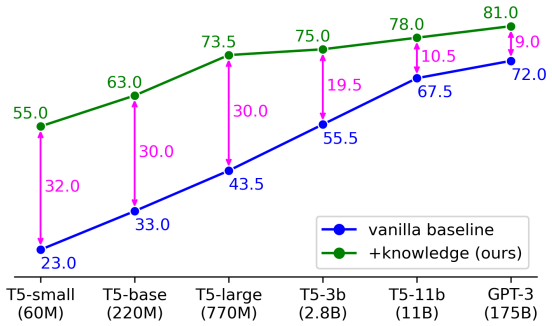

Figure 3: Improvement on top of different sizes of inference model. Measured on Numersense dev set.

with the quantity of knowledge statements. It saturates at $M = 20$ and begins to decline when more knowledge statements are introduced, which may be because more noisy knowledge is generated.

The knowledge integration method. In addition to the knowledge integration method described in §2.2, we experiment with two alternatives: Mixture-of-Experts (MoE) and Product-of-Experts (PoE) (Hinton, 2002). These make the following modifications to Equation 1, respectively:

$$\text{MoE: } p_I(a|q, K_q) \propto \sum_{0 \le m \le M} p_I(a|q_m), \quad (2)$$

$$\text{PoE: } p_I(a|q, K_q) \propto \prod_{0 \le m \le M} p_I(a|q_m). \quad (3)$$

The results in Table 4 indicate that our knowledge integration method – i.e. adaptively choosing the best knowledge to rely on – is best among the three.

Lightweight models and amplification. We found that the size of inference model affects the magnitude of improvement. Figure 3 shows the NumerSense performance gain on top of different sizes of inference model. As we use smaller inference models, the performance gain increases drastically. In particular, with our method the smallest T5 model is as powerful as the T5-3b baseline, and T5-large outperforms the GPT-3 baseline. This indicates that model-generated knowledge can enable high performing, yet lightweight, inference models. Furthermore, the improvement does not diminish as the inference model becomes as big as the knowledge generation model, as the inference by GPT-3 can benefit by 9.0% from the knowledge elicited from itself. This indicates that our method can somewhat *amplify* the useful knowledge already possessed by the model, leading to better predictions.

4.4 Human Evaluation

We conduct a human evaluation on NumerSense and QASC to study the quality of generated knowledge and the interpretability of its impact on task performance.

Evaluation. We report the quality of knowledge statements along four axes: (1) *Grammaticality*: whether it is grammatical; (2) *Relevance*: whether it is relevant to the topic or concepts mentioned on the question; (3) *Factuality*: whether it is (mostly) factually correct; and (4) *Helpfulness*: whether it helps answering the question in an either direct or indirect way, and may fall into one of the three categories: helpful (i.e. supports the correct answer), harmful (i.e. negates the correct answer or supports an incorrect answer), or neutral (neither helpful nor harmful). These metrics are adapted from Shwartz et al. (2020) and are defined in Appendix A.3.

From each dataset, we sample up to 50 *selected knowledge* (§2.2) that change the correctness of T5-11b's prediction (i.e. rectifies model prediction from wrong to right, or misleads model prediction from right to wrong). The knowledge are labeled by two NLP experts and a moderate level of agreement was reached (Fleiss Kappa $\kappa = 0.57$ (Landis and Koch, 1977)). To ensure objectivity, it is not revealed to the annotators whether the knowledge rectifies or misleads the model prediction.

Results. Figure 4 summarizes the results. The vast majority of selected knowledge are grammatical and relevant to the question, and 83% of them are factually correct. 72% are seen as being helpful for answering the question according the human evaluators, whereas 13% are harmful. Out of the knowledge statements that rectify the model predictions, 93% are labeled as helpful by the human evaluators; in contrast, when the knowledge statement misleads the model, only 21% are labeled as helpful, and 39% harmful. Of the knowledge deemed helpful by human *and* rectifies model prediction, 95% are factual, while of those deemed harmful by human *and* misleads model prediction, 86% are non-factual, suggesting that improving

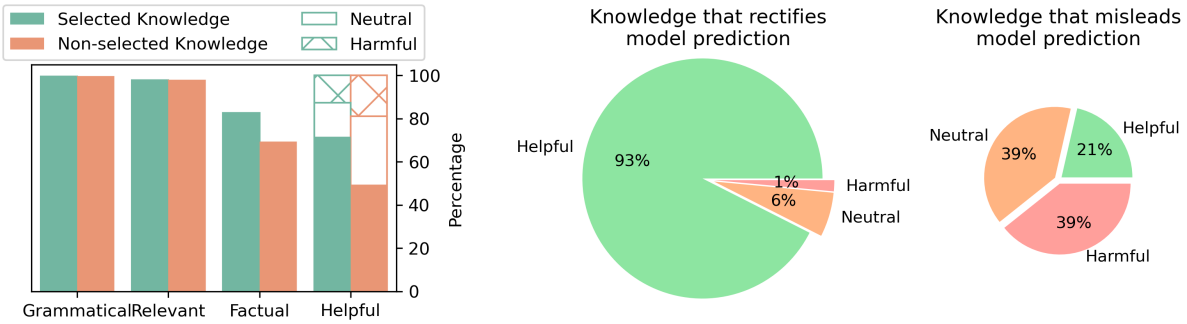

Figure 4: Human evaluation of generated knowledge. **Left:** Percentage of good knowledge statements along each axis. **Right:** Agreement between human and machine on helpfulness of selected knowledge.

knowledge factuality is a promising path towards more helpful knowledge. We also analyzed the non-selected knowledge and found that these statements have slightly lower factuality and helpfulness than the selected knowledge.

4.5 Qualitative Examples

Table 5 shows a few examples where the generated knowledge rectifies model prediction. Due to space constraints we only show the *selected knowledge* (§2.2) for each question. In all examples, the model without prompted knowledge assigns a higher score to an incorrect answer than the correct answer, while with knowledge prompting, the correct answer is assigned a much higher score. Prompting with generated knowledge can transform commonsense reasoning into explicit reasoning procedures such as paraphrasing, induction, deduction, analogy, abductive reasoning, logical elimination, negation, and numerical reasoning.

5 Related Work

Knowledge can be elicited from pretrained language models. Numerous works have shown that pretrained language models implicitly contain large a amount of knowledge that can be queried via conditional generation (Davison et al., 2019; Petroni et al., 2019; Jiang et al., 2020). Consequently, these models can directly perform inference on tasks like commonsense reasoning (Trinh and Le, 2018; Yang et al., 2020), text classification (Shin et al., 2020; Puri and Catanzaro, 2019), and natural language inference (Shin et al., 2020; Schick and Schütze, 2021). Inspired by these observations, we elicit question-related knowledge in an explicit form from language models and use them to guide the inference.

Leveraging external knowledge for commonsense reasoning. Some work uses external commonsense knowledge bases to make improvements on various NLP tasks, including commonsense reasoning. One approach is to inject commonsense knowledge into language models, either by pretraining on knowledge bases (Ma et al., 2021; Chang et al., 2020; Mitra et al., 2019; Zhong et al., 2019) or finetuning the model so that it can reason with additional retrieved knowledge (Chang et al., 2020; Mitra et al., 2019; Bian et al., 2021). Another direction is to ground the question into a knowledge graph and do inference with graph-based reasoning (Lin et al., 2019; Lv et al., 2020; Yasunaga et al., 2021).

A common prerequisite of these methods is a high-quality, high-coverage, in-domain commonsense knowledge base (Ma et al., 2019). Some commonsense reasoning datasets are derived from existing knowledge bases; for example, CommonsenseQA (Talmor et al., 2019) is derived from ConceptNet (Speer et al., 2017), and Social IQA (Sap et al., 2019b) is derived from ATOMIC (Sap et al., 2019a). For such datasets, it is natural to elicit related knowledge from the underlying knowledge base that derived them, and typically this would demonstrate considerable gains (Mitra et al., 2019; Chang et al., 2020). However, if there is a domain mismatch between the dataset and the knowledge base, such gains tend to diminish (Mitra et al., 2019; Ma et al., 2019). This becomes a bottleneck when encountering datasets that have no suitable knowledge base (e.g. NumerSense (Lin et al., 2020) and CommonsenseQA 2.0 (Talmor et al., 2021)), or when the system needs to handle commonsense queries that do not fit in any of the commonsense domains represented by an existing knowledge base. Our work overcomes this diffi-

Dataset	Question / *Knowledge*	Prediction	Score	Reasoning
NumerSense	**clams have evolved to have [M] shells.** *Clams have a bivalve shell.*	no two	0.37 | 0.18 0.89	Commonsense Paraphrasing
NumerSense	**an easel can have [M] or four legs.** *A tripod is a kind of easel.*	two three	0.45 | 0.45 0.46	Commonsense Induction
CSQA	**Where does a heifer's master live?** *The master of a heifer is a farmer.*	slaughter house farm house	0.89 | 0.01 0.92	Commonsense Deduction
CSQA	**Aside from water and nourishment what does your dog need?** *Dogs need attention and affection.*	walked lots of attention	0.55 | 0.04 0.91	Commonsense Elimination
CSQA	**I did not need a servant. I was not a what?** *People who have servants are rich.*	in charge rich person	0.47 | 0.32 0.99	Commonsense Abduction
CSQA2	**Part of golf is trying to get a higher point total than others.** *The player with the lowest score wins.*	yes no	1.00 | 0.00 1.00	Commonsense Negation
CSQA2	**Eighth plus eight is smaller than fifteen.** *Eighth plus eight is sixteen, which is larger than fifteen.*	yes no	0.97 | 0.03 1.00	Commonsense Numerical
QASC	**[M] is used for transportation.** *Bicycles are used for transportation.*	plastic boats	0.41 | 0.12 0.74	Commonsense Analogy

Table 5: More examples where prompting with generated knowledge reduces the reasoning type and rectifies the prediction. The first row of each section is the original question and the inference results associated with it; the second row is a model-generated knowledge statement that prompts the inference model. We show correct answers in green, incorrect answers in red, and their corresponding scores assigned by the inference model.

culty by leveraging pretrained language models as the source of commonsense knowledge.

Adding generated text during inference. Recently, several works show that model performance on commonsense reasoning can be boosted by augmenting the question with model-generated text, such as clarifications, explanations, and implications. Self-talk (Shwartz et al., 2020) elicits clarifications to concepts in the question and appends them to the inference model input. Contrastive explanations (Paranjape et al., 2021) prompts inference models with generated explanations that contrast between two answer choices. The aforementioned methods depend on task-specific templates to inquire the generator, which means they are only capable of eliciting a limited variety of knowledge and require careful hand-crafting to transfer to new tasks. Other explanation-based methods (Latcinnik and Berant, 2020; Rajani et al., 2019) finetune the generator model so that it produces explanations that are used for question augmentation. DynaGen (Bosselut et al., 2021) uses pretrained commonsense models to generate implications of a question and expands the inference input with these generations. However, its usage of COMeT (Bosselut et al., 2019) as the generator confines its applicability to the social commonsense domain. Our work contributes to this general line of research, yet different from these previous methods that elicit knowledge with task-specific templates or from finetuned knowledge generators, our method requires only a few human-written demonstrations in the style of the task, making it much more flexible, easy-to-transfer, and engineering-efficient.

6 Conclusion

We introduce generated knowledge prompting, a simple method to elicit and integrate knowledge from language models so as to improve performance on commonsense reasoning tasks. In particular, we generate knowledge statements by prompting a language model with task-specific, human-written, few-shot demonstrations of question-knowledge pairs. We show that knowledge can be integrated by simply plugging it in at inference time, with no need to finetune the model for knowledge integration. Our method shows effectiveness across multiple datasets, sets the new state-of-the-art on three commonsense reasoning tasks, and works under a variety of settings. The method's success highlights language models as sources of flexible, high-quality knowledge for commonsense reasoning.

References

Ning Bian, Xianpei Han, Bo Chen, and Le Sun. 2021. Benchmarking knowledge-enhanced commonsense question answering via knowledge-to-text transformation. arXiv preprint arXiv:2101.00760.

Antoine Bosselut, Ronan Le Bras, and Yejin Choi. 2021. Dynamic neuro-symbolic knowledge graph construction for zero-shot commonsense question answering. In Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI).

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.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165.

Ting-Yun Chang, Yang Liu, Karthik Gopalakrishnan, Behnam Hedayatnia, Pei Zhou, and Dilek Hakkani-Tur. 2020. Incorporating commonsense knowledge graph in pretrained models for social commonsense tasks. In Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 74–79, Online. Association for Computational Linguistics.

Joe Davison, Joshua Feldman, and Alexander Rush. 2019. Commonsense knowledge mining from pretrained models. 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 1173–1178, Hong Kong, China. Association for Computational Linguistics.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. Neural computation, 14(8):1771–1800.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. arXiv preprint arXiv:1904.09751.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2020. How can we know what language models know? Transactions of the Association for Computational Linguistics, 8:423–438.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 1896–1907, Online. Association for Computational Linguistics.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 8082–8090.

J Richard Landis and Gary G Koch. 1977. The measurement of observer agreement for categorical data. biometrics, pages 159–174.

Veronica Latcinnik and Jonathan Berant. 2020. Explaining question answering models through text generation. arXiv preprint arXiv:2004.05569.

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, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! NumerSense: Probing Numerical Commonsense Knowledge of Pre-Trained Language Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 6862–6868, Online. Association for Computational Linguistics.

Nicholas Lourie, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Unicorn on rainbow: A universal commonsense reasoning model on a new multitask benchmark. arXiv preprint arXiv:2103.13009.

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.

Kaixin Ma, Jonathan Francis, Quanyang Lu, Eric Nyberg, and Alessandro Oltramari. 2019. Towards generalizable neuro-symbolic systems for commonsense question answering. In Proceedings of the First Workshop on Commonsense Inference in Natural Language Processing, pages 22–32, Hong Kong, China. Association for Computational Linguistics.

Kaixin Ma, Filip Ilievski, Jonathan Francis, Yonatan Bisk, Eric Nyberg, and Alessandro Oltramari. 2021. Knowledge-driven data construction for zero-shot evaluation in commonsense question answering. In 35th AAAI Conference on Artificial Intelligence.

Arindam Mitra, Pratyay Banerjee, Kuntal Kumar Pal, Swaroop Mishra, and Chitta Baral. 2019. How additional knowledge can improve natural language commonsense question answering? arXiv preprint arXiv:1909.08855.

Bhargavi Paranjape, Julian Michael, Marjan Ghazvininejad, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021. Prompting contrastive explanations for commonsense reasoning tasks. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 4179–4192, Online. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. arXiv preprint arXiv:1912.10165.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4932–4942, Florence, Italy. Association for Computational Linguistics.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019a. Atomic: An atlas of machine commonsense for if-then reasoning. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 33, pages 3027–3035.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019b. 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.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pages 255–269, Online. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4222–4235, Online. Association for Computational Linguistics.

Vered Shwartz, Peter West, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Unsupervised commonsense question answering with self-talk. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4615–4629, Online. Association for Computational Linguistics.

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. Commonsenseqa 2.0: Exposing the limits of ai through gamification.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. arXiv preprint arXiv:1806.02847.

Jun Yan. 2021. Usc ink submission on numersense.

Jheng-Hong Yang, Sheng-Chieh Lin, Rodrigo Nogueira, Ming-Feng Tsai, Chuan-Ju Wang, and Jimmy Lin. 2020. Designing templates for eliciting commonsense knowledge from pretrained sequence-to-sequence models. In Proceedings of the 28th International Conference on Computational Linguistics, pages 3449–3453, Barcelona, Spain (Online). International Committee on 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.

Yuhui Zhang. 2021. Stanford submission on numersense.

Wanjun Zhong, Duyu Tang, Nan Duan, Ming Zhou, Jiahai Wang, and Jian Yin. 2019. Improving question answering by commonsense-based pre-training. In CCF International Conference on Natural Language Processing and Chinese Computing, pages 16–28. Springer.

A Appendix

A.1 Comparison with Prior Methods

Table 6 summarizes the comparison between our generated knowledge prompting method and prior methods that add generated text to an inference model for commonsense reasoning tasks. Our method is unique because it uses few-shot demonstrations to prompt for knowledge generation, and can apply to finetuned inference models without joint finetuning with knowledge.

A.2 Prompts for Knowledge Generation

Table 7 through 10 shows the full prompts for knowledge generation that we use for each evaluated task: NumerSense, CSQA, CSQA2, and QASC.

A.3 Human Evaluation Guidelines

Table 11 and 12 shows the detailed guidelines we use for human evaluation of generated knowledge.

Method	Knowledge Generator	Inference Model
CAGE (Rajani et al., 2019)	task-finetuned	joint-finetuned
Latcinnik and Berant (2020)	task-finetuned	joint-finetuned
DynaGen (Bosselut et al., 2021)	task-finetuned	joint-finetuned
Self-talk (Shwartz et al., 2020)	template-prompted	0-shot
Contrastive expl. (Paranjape et al., 2021)	template-prompted	0-shot & joint-finetuned
Generated knowledge prompting (ours)	demonstrations-prompted	0-shot & task-finetuned

Table 6: Comparison of methods that add generated text to an inference model. **Knowledge Generator**: *task-finetuned* – a model finetuned to generate task-specific knowledge; *template-prompted* – an off-the-shelf LM from which knowledge statements are elicited via templates; *demonstration-prompted* – an off-the-shelf LM from which knowledge statements are elicited via few-shot demonstrations (§2.1). **Inference Model**: *0-shot* – an off-the-shelf LM that is set up to make predictions; *task-finetuned* – a model finetuned with task training data (and without seeing extra knowledge); *joint-finetuned* – a model finetuned with task training data *and* generated knowledge.

Task	Prompt
NumerSense	Generate some numerical facts about objects. Examples: Input: **penguins have <mask> wings.** Knowledge: *Birds have two wings. Penguin is a kind of bird.* Input: **a parallelogram has <mask> sides.** Knowledge: *A rectangular is a parallelogram. A square is a parallelogram.* Input: **there are <mask> feet in a yard.** Knowledge: *A yard is three feet.* Input: **water can exist in <mask> states.** Knowledge: *There states for matter are solid, liquid, and gas.* Input: **a typical human being has <mask> limbs.** Knowledge: *Human has two arms and two legs.* Input: **{question}** Knowledge:

Table 7: Prompt for knowledge generation on NumerSense. Demonstration examples are manually written and the knowledge enables explicit reasoning procedures to answer the input question.

Task	Prompt
CSQA	Generate some knowledge about the concepts in the input. Examples: Input: **Google Maps and other highway and street GPS services have replaced what?** Knowledge: *Electronic maps are the modern version of paper atlas.* Input: **The fox walked from the city into the forest, what was it looking for?** Knowledge: *Natural habitats are usually away from cities.* Input: **You can share files with someone if you have a connection to a what?** Knowledge: *Files can be shared over the Internet.* Input: **Too many people want exotic snakes. The demand is driving what to carry them?** Knowledge: *Some people raise snakes as pets.* Input: **The body guard was good at his duties, he made the person who hired him what?** Knowledge: *The job of body guards is to ensure the safety and security of the employer.* Input: **{question}** Knowledge:

Table 8: Prompt for knowledge generation on CSQA. Demonstration examples are selected from the CSQA training set; we manually write relevant knowledge for each input question.

Task	Prompt
CSQA2	Generate some knowledge about the input. Examples: Input: **Greece is larger than mexico.** Knowledge: *Greece is approximately 131,957 sq km, while Mexico is approximately 1,964,375 sq km, making Mexico 1,389% larger than Greece.* Input: **Glasses always fog up.** Knowledge: *Condensation occurs on eyeglass lenses when water vapor from your sweat, breath, and ambient humidity lands on a cold surface, cools, and then changes into tiny drops of liquid, forming a film that you see as fog. Your lenses will be relatively cool compared to your breath, especially when the outside air is cold.* Input: **A fish is capable of thinking.** Knowledge: *Fish are more intelligent than they appear. In many areas, such as memory, their cognitive powers match or exceed those of 'higher' vertebrates including non-human primates. Fish's long-term memories help them keep track of complex social relationships.* Input: **A common effect of smoking lots of cigarettes in one's lifetime is a higher than normal chance of getting lung cancer.** Knowledge: *Those who consistently averaged less than one cigarette per day over their lifetime had nine times the risk of dying from lung cancer than never smokers. Among people who smoked between one and 10 cigarettes per day, the risk of dying from lung cancer was nearly 12 times higher than that of never smokers.* Input: **A rock is the same size as a pebble.** Knowledge: *A pebble is a clast of rock with a particle size of 4 to 64 millimetres based on the Udden-Wentworth scale of sedimentology. Pebbles are generally considered larger than granules (2 to 4 millimetres diameter) and smaller than cobbles (64 to 256 millimetres diameter).* Input: **{question}** Knowledge:

Table 9: Prompt for knowledge generation on CSQA2. Demonstration examples are selected from the CSQA2 training set; we use the annotated *Google featured snippet* as the knowledge.

Task	Prompt
QASC	Generate some knowledge about the input. Examples: Input: **What type of water formation is formed by clouds?** Knowledge: *Clouds are made of water vapor.* Input: **What can prevent food spoilage?** Knowledge: *Dehydrating food is used for preserving food.* Input: **The process by which genes are passed is** Knowledge: *Genes are passed from parent to offspring.* Input: **The stomach does what in the body?** Knowledge: *The stomach is part of the digestive system.* Input: **What can cause rocks to break down?** Knowledge: *Mechanical weathering is when rocks are broken down by mechanical means.* Input: **{question}** Knowledge:

Table 10: Prompt for knowledge generation on QASC. Demonstration examples are selected from the QASC training set; we use one of the gold separate facts as the knowledge.

Attribute	Options	Description and Examples
Grammaticality	grammarical; ungrammatical but understandable; completely gibberish	Whether the knowledge statement is grammatical. We are aware that some of the statements are not fully grammatical. If you can still understand what the statement says, even if it's an incomplete sentence or slightly ungrammatical, please select the "ungrammatical but understandable" option.
Relevance	relevant; not relevant	Whether a knowledge statement is relevant to the given question. A statement is relevant if it covers one the same topics as the question, or contains a salient concept that is same or similar to one in the question. Examples: [Question] you may take the subway back and forth to work <mask> days a week. [Knowledge] You take the subway back and forth to work five days a week. [Judgment] Relevant, because the question and knowledge are both about the topic of business days. [Question] a bradypus torquatus is native to brazil and has <mask> toes on each limb. [Knowledge] A bradypus torquatus is a kind of mammal. A mammal has four limbs. [Judgment] Relevant, because the question and knowledge share a common salient concept "bradypus torquatus".
Factuality	factual; not factual	Whether a knowledge statement is (mostly) factually correct or not. If there are exceptions or corner cases, it can still be considered factual if they are rare or unlikely. Examples: [Knowledge] A limousine has four doors. [Judgment] Factual. [Knowledge] A human hand has four fingers and a thumb. [Judgment] Factual, despite that there are exceptions – people with disabilities may have less or more fingers. [Knowledge] A rectangle is a shape with two equal sides. [Judgment] Not factual, because a rectangle has four sides.

Table 11: Human evaluation guidelines. Continued in Table 12.

Attribute	Options	Description and Examples
Helpfulness	helpful; neutral; harmful	Whether a knowledge statement provides useful information in support OR contradiction of the answer. A statement is helpful if it supports the answer either directly or indirectly. More on indirect support – The statement might not directly answer the question directly, yet it may support an indirect reasoning path that reaches the answer. A statement is harmful if it negates the answer or supports an alternative potential answer either directly or indirectly. A statement is neutral if it is neither helpful nor harmful. Examples: [Question] you may take the subway back and forth to work <mask> days a week. [Answer] five [Knowledge] You take the subway back and forth to work five days a week. [Judgment] Helpful. Because the statement directly supports the answer. [Question] spiders have <mask> legs. [Answer] eight [Knowledge] Arachnids have eight legs. [Judgment] Helpful. Although the statement does not directly refer to spiders, together with the fact that "spiders are a kind of arachnids" it completes a reasoning chain in deriving the answer. [Question] a game of chess may have <mask> outcomes. [Answer] three [Knowledge] A game of chess has two outcomes. [Judgment] Harmful. Since the statement supports answering "two" instead of "three". [Question] a bradypus torquatus is native to brazil and has <mask> toes on each limb. [Answer] three [Knowledge] A bradypus torquatus is a kind of mammal. A mammal has four limbs. [Judgment] Neutral. The statement does not provide information in favor or contrast of the answer.

Table 12: (continued) Human evaluation guidelines.