CIKQA: Learning Commonsense Inference with a Unified Knowledge-in-the-loop QA Paradigm

Hongming Zhang[1,2]**, Yintong Huo**[3]***, Yanai Elazar**[4,5]**,**
Yangqiu Song[1]**, Yoav Goldberg**[4,5]**, Dan Roth**[2]
[1]HKUST, [2]UPenn, [3]CUHK, [4]Bar Ilan University, [5]AI2
{hzhangal,yqsong}@cse.ust.hk, ythuo@cse.cuhk.edu.hk
{yanaiela,yoav.goldberg}@gmail.com, danroth@seas.upenn.edu

Abstract

We propose a new commonsense reasoning benchmark to motivate commonsense reasoning progress from two perspectives: (1) Evaluating whether models can distinguish knowledge quality by predicting if the knowledge is enough to answer the question; (2) Evaluating whether models can develop commonsense inference capabilities that generalize across tasks. We first extract supporting knowledge for each question and ask humans to annotate whether the auto-extracted knowledge is enough to answer the question or not. After that, we convert different tasks into a unified question-answering format to evaluate the models' generalization capabilities. We name the benchmark Commonsense Inference with Knowledge-in-the-loop Question Answering (**CIKQA**). Experiments show that with our learning paradigm, models demonstrate encouraging generalization capabilities. At the same time, we also notice that distinguishing knowledge quality remains challenging for current commonsense reasoning models.

1 Introduction

Understanding human language requires both language knowledge (e.g., grammar and semantics) and world knowledge, which can be further divided into factual and commonsense knowledge (Katz and Fodor, 1963). Recently, the community has made great progress in helping machines acquire and apply language and factual knowledge. However, how to help machines acquire and infer over commonsense is still unclear. To answer this question, many commonsense reasoning datasets (Roemmele et al., 2011; Sakaguchi et al., 2020; Talmor et al., 2019; Zellers et al., 2019; Lin et al., 2020) have been proposed. Even though they target different knowledge types, modalities, and formats, they often

follow a standard supervised learning setting that aims at helping machines solve a specific task with training data. However, two limitations of this learning paradigm have restricted the development of commonsense reasoning systems.

First, there is no clear separation between knowledge and inference. As discussed in Elazar et al. (2021), a common phenomenon is that larger training data will lead to better performance, mainly because richer knowledge is covered. However, due to the large scale of commonsense knowledge, it is infeasible to annotate a large enough training set for each task, and the responsibility of the training data should be teaching models how to make inferences rather than acquire commonsense knowledge. Several recent works have explored using structured knowledge for commonsense reasoning tasks (Lin et al., 2019; Lv et al., 2020; Paul and Frank, 2020). However, as these works did not clearly analyze the coverage of the structured knowledge (i.e., knowledge graphs (KGs)), it is still unclear what the performance means, better knowledge coverage, or better inference capability. To investigate what is behind this learning process, we propose to equip each question with auto-extracted knowledge and ask humans to annotate whether the knowledge is sufficient to answer the question. By doing so, we could evaluate whether models can know if the provided knowledge is good or not and how well they can conduct inference over the provided knowledge to solve the task.

Second, supervised learning may force the model to learn the distribution of the training data rather than a universal inference model. As a result, the model may perform well on the test set that follows the same distribution but fail to generalize (Kejriwal and Shen, 2020). Previously, as different tasks have different formats, it is hard to evaluate the generalization ability of commonsense reasoning models. Following the trend of

---

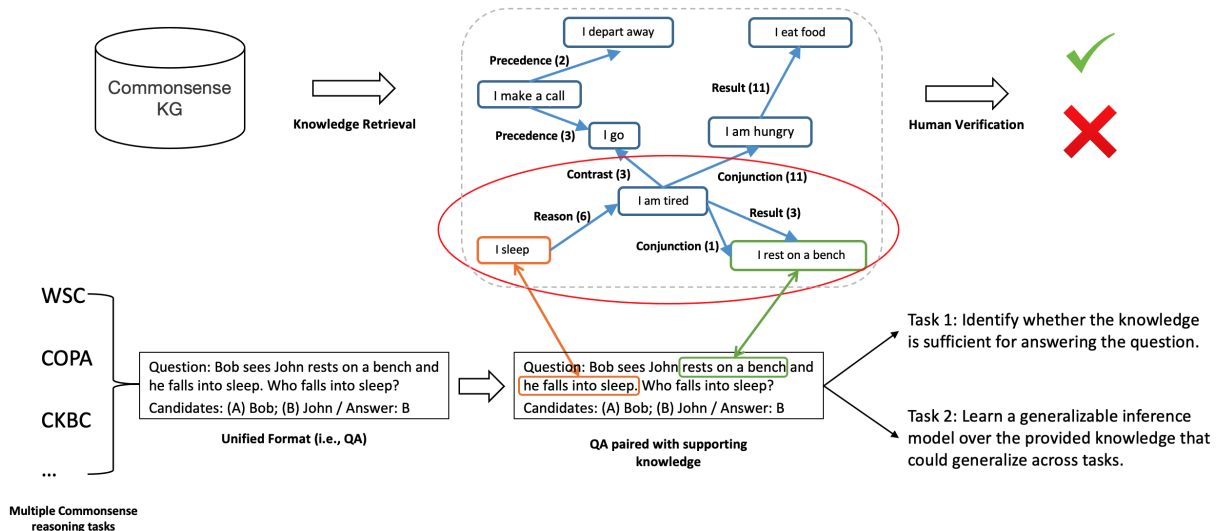* This work was done when the second author was visiting HKUST.

Figure 1: **CIKQA** demonstration. All tasks are converted into a unified format such that we can easily evaluate the generalization capability of all models. We also equip all questions with auto-extracted knowledge graphs from existing KGs and ask humans to annotate whether the knowledge is gold or not. In this example, we expect models to first identify the quality of the knowledge and then conduct inference over the knowledge to solve the question.

using a unified format (i.e., question answering) for different tasks (Khashabi et al., 2020), we propose to convert various commonsense reasoning tasks into a unified QA format such that we can easily and fairly evaluate the generalization ability of learned commonsense reasoning models.

Combining these two lines of effort, we propose a new commonsense inference benchmark Commonsense Inference with Knowledge-in-the-loop QA (**CIKQA**). An example is shown in Figure 1. We first convert several popular commonsense reasoning tasks into a unified QA format and equip them with the relevant knowledge from existing commonsense knowledge graphs. We leverage human annotation to label whether the provided knowledge is correct and enough[1] to answer the question. The **CIKQA** benchmark can motivate us to answer two questions: (1) Whether current models can distinguish the knowledge is gold or not; (2) Can current commonsense inference models generalize across different commonsense reasoning tasks?

Experiments with several recent knowledge-based commonsense reasoning models show that even though current deep models could learn to conduct simple inferences after training with a few examples when gold knowledge is provided, they still cannot learn to distinguish gold knowledge very well. Moreover, although current mod-

els demonstrate encouraging generalization ability across the three tasks we consider, they still struggle with complex inference (e.g., abductive reasoning). We hope that our benchmark[2] can motivate more advanced commonsense inference methods in the future.

2 Dataset Construction

In **CIKQA**, to encourage a generalizable commonsense inference model, we follow previous work (Khashabi et al., 2020; Cohen et al., 2020; Wu et al., 2020; Du and Cardie, 2020) to unify all selected tasks as a binary question answering problem, and equip each question with a supporting knowledge graph G retrieved from existing commonsense KGs. We leverage crowd-sourcing workers to annotate whether the knowledge is gold (i.e., accurate and enough) for answering the question. With that, we can evaluate whether models know how to distinguish gold and knowledge and whether they can learn the generalizable inference with the help of the knowledge. In total, **CIKQA** contains 15 thousand instances from four kinds of commonsense reasoning tasks. Details about task selection, format unification, knowledge extraction, and annotation are as follows.

2.1 Task Selection

In **CIKQA**, we select the following four popular commonsense reasoning tasks:

---

[1] In the rest of the paper, we denote such knowledge as the gold knowledge.

[2] Available at https://github.com/CogComp/CIKQA.

Task Name	Original Assertion	Transformed Question	Answer
HardPCR	The fish ate the worm. It was hungry.	The fish ate the worm. It was hungry. What was hungry?	(A) Fish; (B) Worm
CommonsenesQA	What is a place that someone can go buy a teddy bear?	What is a place where someone can go buy a teddy bear?	(A) Toy store; (B) Shelf
COPA	I drank from the water fountain.	I drank from the water fountain. What was the cause of this?	(A) I was thirsty.; (B) I felt nauseous.
ATOMIC	PersonX buys the bike.	Before PersonX buys the bike, what did PersonX want?	(A) To be social.; (B) To have transportation.

Table 1: Demonstration of the original assertion, transformed questions, and answers. Correct and wrong answers are indicated in blue and red, respectively.

1. HardPCR (Zhang et al., 2021): The hard pronoun coreference resolution (HardPCR) task is one of the most famous commonsense reasoning tasks. For each question, a target pronoun and two candidate mentions are provided, and the task is to select the correct mention that the pronoun refers to. Careful expert annotations are conducted to get rid of the influence of all simple linguistic rules, and the models are required to solve the problem with commonsense reasoning. We include instances from WSC (Levesque et al., 2012), DPR (Rahman and Ng, 2012), and WinoGrande (Sakaguchi et al., 2020). To create a question regarding the target pronoun, we first find the sentence that contains the target pronoun and then determine whether the participating pronoun refers to a person or an object.

2. CommonsenseQA (Talmor et al., 2019) is a multiple-choice question answering dataset. For each question-answer pair, four relevant but wrong concepts are used as the other candidates, and the models are required to select the correct one out of five candidates. In **CIKQA**, we randomly sample a negative answer to make it a binary choice task, which is consistent with other datasets.

3. COPA (Roemmele et al., 2011) focuses on evaluating the understanding of event causality. Two follow-up events are provided for a target event, and models are asked to predict the one caused by or the reason for the target event.

4. ATOMIC (Sap et al., 2019): is a commonsense knowledge graph, which we convert into a completion problem. Given a head concept (e.g., "eat food") and a relation (e.g., "cause"), we want to predict the tail concept, focusing on predicting the edges of ATOMIC.

In COPA and ATOMIC, where the task is to predict the relations between two events or states (e.g., "PersonX eats"-*Causes*-"PersonX is full"), for each triplet, we randomly sample another event or state as the negative tail and ask the model to select the correct one. To make the task challenging and avoid sampling irrelevant events or states, we restrict the sampled negative event or state to be connected with the head of a different triplet (e.g., "PersonX is hungry" from the triplet "PersonX eats"-*CausedBy*-"PersonX is hungry"). For each relation, we write a pattern to generate the question. For example, for the "Causes" relation, we will ask "What can be caused by the event 'PersonX eats'?". Examples of instances in the original datasets and their transformed questions and candidate answers are presented in Table 1.

2.2 Supporting Knowledge Extraction

As discussed in Section 1, a limitation of existing commonsense reasoning benchmarks is that there is no clear boundary between knowledge and inference. As such, it is unclear what is learned from the training data, the knowledge, how to perform inference, or a combination of both. We propose to equip each question with supporting knowledge to address this issue and encourage models to learn inference rather than knowledge from the training data. The question is selected as part of the dataset only if we find supporting knowledge to answer the question. Note that this procedure serves as an improved evaluation setup than purely supervised learning and not as a solution to commonsense reasoning. This section introduces the selected commonsense knowledge graphs and then introduces how we extract the corresponding commonsense knowledge for each question.

2.2.1 Commonsense KG Selection

Many commonsense knowledge graphs were developed to enhance machines' commonsense reasoning abilities, including ConceptNet (Liu and Singh, 2004), ATOMIC (Sap et al., 2019), GLUCOSE (Mostafazadeh et al., 2020), and ASER (Zhang et al., 2020a). Among these four, ConceptNet, ATOMIC, and GLUCOSE were constructed via crowd-sourcing, while ASER was constructed automatically with information extraction techniques. Besides ATOMIC, which is used as one of the tasks, we use the other KBs as supporting knowledge resources.

2.2.2 Supporting Graph Extraction

Here we introduce how to extract the supporting knowledge from external commonsense knowledge bases. For each question, we need to obtain a sub-graph from supporting knowledge graphs to contain the relevant commonsense knowledge about the question. The sub-graph extraction process includes the following three steps: (1) Pre-processing: Convert each question into several key sentences; (2) Matching: Match the sentences into nodes in the KG; (3) Extraction: Retrieve the relevant sub-graphs from the entire KG. In what follows, we give some more details on each of the steps.

Data Pre-processing: For each question and the associated candidate answers, we first replace the question words (e.g., "What") with the two candidate answers such that they become two declarative sentences. For instance, if the question is "The fish ate the worm. It was hungry. Who is hungry?" and the candidates are "Fish" and "Worm," we will convert the question into the declarative sentence: "The fish is hungry" and "The worm is hungry." As a result, we will get three sentences for this question: "The fish ate the worm," "The fish is hungry," and "The worm is hungry."

KG Matching: After getting the declarative sentences containing the question and key answers, we map them to nodes in knowledge graphs to extract the relevant knowledge. Considering that each sentence may have multiple words and it is often hard to find an exact match, we adopt an embedding-based fuzzy matching technique. For each sentence and node in the KG, we treat them as a sentence and get the corresponding representations with SimCSE (Gao et al., 2021). For each input sentence, SimCSE encodes the sentence into a vector. A close distance between two

vectors indicates that the two sentences are similar to each other. We use cosine similarity on the obtained representations to measure the similarity between two sentences.[3] Since there are 287 thousand nodes in GLUCOSE and 194 million nodes in ASER, it is computationally infeasible to compute the cosine similarity between sentences pair by pair. Thus we use an approximation. For each extracted sentence, we first apply Faiss (Johnson et al., 2017), a large-scale similarity-based matching algorithm that first clusters all KG nodes in the vector space to increase the matching efficiency when finding the top N nodes in the KG. We encode all the nodes of the graph and index them using Faiss (Johnson et al., 2017). Then, we can perform fast and quick retrieval of the most-similar nodes with each query sentence. After that, we sort the N nodes based on the cosine similarity to find the top K similar nodes. We set N and K to be 60 and 1, respectively. On average, it takes 25 seconds to retrieve the relevant nodes for each question.

Graph Extraction: Next, we extract the sub-graph that contains all the relevant nodes. We denote the extracted m nodes as $n_1, n_2, ..., n_m$, and for each of them, we find K similar nodes from the KG. The resulting matched node sets are denoted as $\mathcal{N}_1, \mathcal{N}_2, ..., \mathcal{N}_m$. For any pair of nodes $n \in \mathcal{N}_i$ and $n' \in \mathcal{N}_j$ ($i \neq j$), if there exists a path in the KG between n and n', we will keep that path. After adding all paths together, we will get the final sub-graph. On average, constructing a graph for each question takes less than two seconds.

Knowledge Quality Annotation: Since our extraction method is automatic, some of the sub-graphs may be irrelevant or insufficient for answering the questions. We use crowdsourcing to annotate whether the extracted knowledge is gold (i.e., accurate and enough), five per example. The average Inter-annotator agreement (Cohen's kappa statistic) is 0.83, indicating our annotation's high quality. In the end, we apply a strict standard (at least four of five annotators need to vote for gold) to select the gold knowledge.

2.3 CIKQA Statistics

We report the dataset statistics in Table 2. In total, **CIKQA** contains 14,599 instances, among which Hard PCR and ATOMIC provide the most

---

[3]We also tried other techniques such as string match, ROUGE (Lin, 2004), and BLEURT (Sellam et al., 2020), but found them to be either inaccurate or too slow for our scale.

Task Name	# Instance by Knowledge Resource			# Total Instance	Avg Sub-graph Size	# Gold Instance
	ASER	ConceptNet	GLUCOSE			
HardPCR	2,030	202	2,143	4,375	2.85	670
CommonsenseQA	530	31	37	598	3.19	59
COPA	103	41	149	293	3.03	78
ATOMIC	5,655	212	3,466	9,333	2.67	2,200
Total	8,318	486	5,795	14,599	2.75	3,007

Table 2: **CIKQA** statistics. "Avg Sub-graph Size" is the average graph size measured by the number of edges. "# Gold Instance" means the number of instances supported by different knowledge resources and annotated gold (i.e., Accurate and Enough) knowledge.

questions because their original datasets are much larger than others. According to the annotation, 20.59% of the instances contain gold knowledge. Based on our analysis, annotators hold a very strict standard for selecting the gold knowledge. We randomly split the dataset into training, development, and testing sets for each task with a standard 8:1:1 splitting. As a result, we get 11,678 training, 1,459 development, and 1,462 testing instances.

3 Experiment Setup

We present the performance of the following commonsense inference models on **CIKQA**:

(1) Vanilla LM: We use the language model (LM) based multiple-choice (MC) model as the basic baseline. For each candidate answer, we concatenate it with the question and feed it to the model. After getting the sentence representation, a linear layer is used to obtain a score and trained with a cross-entropy loss.

(2) KagNet: As one of the pioneering works that utilized structured knowledge for solving commonsense reasoning tasks, KagNet (Lin et al., 2019) first uses a graph convolution network to encode the knowledge graph and then apply an LSTM based hierarchical attention mechanism to encode the knowledge paths that start with the nodes corresponding to the question and end with nodes corresponding to the answer. At the same time, KagNet encodes the question and answers with pre-trained LMs. In the end, it concatenates all representations for the final prediction.

(3) Graph-Based Reasoning (GBR): Instead of only encoding paths starting with the question nodes and ending with answer nodes, in GBR (Lv et al., 2020), they propose to run a depth-first algorithm over the knowledge graph to generate a sequence of paths as the supporting knowledge paths.

(4) Multi-Head Knowledge Attention (MHKA):

To further utilize the knowledge, MHKA (Paul and Frank, 2020) uses a transformer network to model the paths from the question nodes and answer nodes, then concatenates the knowledge and context representation for the final prediction.

(5) Graph-to-Text (G2T): In the end, we also evaluate a simple yet effective approach of combining structured knowledge and language models: Graph-to-Text (Bian et al., 2021), which first verbalizes knowledge into a sentence and then concatenates the knowledge sentence and target question together. On top of that, a transformer-based model is used to encode the input sentence and make the final prediction.

Implementation Details We implement all experiments with Huggingface (Wolf et al., 2019). We select BERT-base (Devlin et al., 2019) as the base language model for all models. The batch size is set to 16. All models are trained for 10,000 steps[4], and the best-performing checkpoints on the dev set are evaluated. For our model, we set both the number of random walk paths and the walk length to five. Considering that the auto-extracted knowledge could contain noise or miss certain knowledge, we add a "gold knowledge" setting, where only examples with the gold knowledge are used for training and testing, for all models as the upper bound of their model. All other hyper-parameters are the same as the base language model. All models are trained with GTX 2080, and the average running time is 12 hours.

4 Result Analysis

We first conduct analysis experiments to evaluate to what extent the provided knowledge could help existing models. For each model, we train it with different numbers of training instances and report the average performance and standard de-
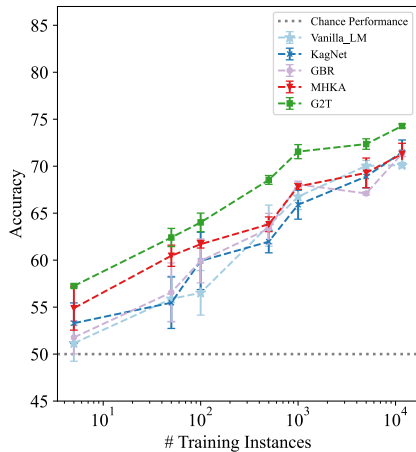
---

[4]All models converge at 10,000 steps.

Figure 2: Learning curves of all evaluated models on all instances of **CIKQA**. We evaluate all models with the full dataset.
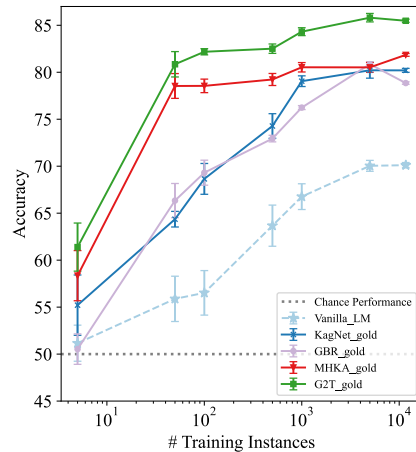

Figure 3: Learning curves of all evaluated models on the gold subset of **CIKQA**, where only instances with gold knowledge are used for training and testing.

viation of five trials. Experiment results with all instances and the gold subset of **CIKQA**, where only instances with gold knowledge are used for training and testing, are presented in Figure 2 and 3, respectively. From the results, we can make the following observations. First, when explicitly including the knowledge, all inference models outperform the baseline model with no supporting knowledge, especially G2T. When the auto-extracted and gold knowledge is provided, G2T outperforms the baseline Vanilla LM model by 4.17 and 15.34 accuracy, respectively. It supports our assumption that learning all knowledge from the limited training data is hard, and external structured knowledge could help. At the same time, we also notice a significant gap between auto-extracted knowledge and gold knowledge. For example, if gold knowledge is available, models could learn to answer the questions with only a few examples. This indicates that the knowledge quality can significantly impact models' performance, which further shows the importance of distinguishing whether the knowledge is gold or not automatically. Last but not least, we can see that G2T outperforms other inference models in most settings, which shows that with the help of current large-scale LMs, jointly encoding questions and knowledge is more efficient and a more effective strategy than acquiring them separately. Due to the simplicity and efficiency of G2T, we will conduct the rest analysis experiments with G2T.

4.1 Distinguishing the Gold Knowledge

Humans can say "I do not know" when they find out that they cannot answer a question with their knowledge. To investigate whether current deep models have a similar capability, we use G2T as an example to test whether these deep models can distinguish the gold knowledge. For each (question, answer, and knowledge) triplet, we train and test G2T with annotated knowledge quality labels. To address the imbalanced distribution problem, we randomly select the same number of "Not Gold" examples as the "Gold" ones to make the dataset balanced. From the results in Figure 4, we can see that the performance of G2T can be improved slightly with the increase of training data. However, after seeing thousands of examples, it still can only achieve 0.65 accuracy on a binary classification problem. It shows that knowing when to say "I do not know" is still a challenging task for current deep models, which is consistent with the observations in previous literature that deep models cannot understand the reasons and knowledge they used to answer questions (Zhang et al., 2020b; Sanh et al., 2022). We hope that **CIKQA** could motivate more future work on this important research problem.

4.2 Generalization Ability

An important assumption and motivation behind the unified problem design of **CIKQA** is that even though the commonsense could be enormous, the inference rules over commonsense knowledge can

Training Task	Testing Task			
	Hard PCR	CommonsenseQA	COPA	ATOMIC
Hard PCR	-	37.50 → 52.30	75.00 → 53.24	44.13 → 53.32
CommonsenseQA	50.00 → 50.14	-	62.50 → 56.67	56.34 → 70.56
COPA	45.95 → 51.26	62.50 → 58.33	-	49.77 → 62.96
ATOMIC	39.19 → 50.76	50.00 → 76.67	62.50 → 73.33	-

(a) Full Dataset (Vanilla LM (without knowledge)→ G2T (with knowledge))

Training Task	Testing Task			
	Hard PCR	CommonsenseQA	COPA	ATOMIC
Hard PCR	-	46.67 → 51.67	63.33 → 56.67	51.85 → 55.78
CommonsenseQA	49.32 → 50.32	-	50.00 → 75.00	60.39 → 91.08
COPA	52.51 → 54.79	56.67 → 87.50	-	53.01 → 76.06
ATOMIC	50.46 → 51.35	68.33 → 93.75	56.67 → 87.50	-

(b) Gold Subset (Vanilla LM (without knowledge)→ G2T (with knowledge))

Table 3: Generalization ability demonstration. We report the performance on both the full dataset and the gold dataset (i.e., only questions with gold knowledge are selected for training and testing) to show the generalization ability. Strong and moderate generalization settings are indicated with the green and orange background, respectively.
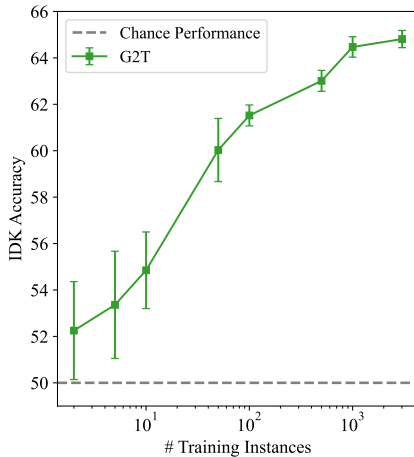

Figure 4: The learning curve of G2T on the gold knowledge identification task.

be limited. As a result, even though we could not learn all the commonsense from limited training data, we can learn how to conduct inference with several tasks and then generalize to others. In this section, we conduct experiments with both the "Without Knowledge" and "With Knowledge" models to show that we can gain such generalization ability across different tasks with our unified formulation. We conduct experiments on two settings: (1) Full Set: We train and test the model with the whole dataset; (2) Gold Subset: We only train and test the model on questions where the supporting graph is annotated as gold. We train the model with questions from a specific task and test it on all tasks. The results are in Table 3.

From the results, we can see that the knowledge can help models to generalize well among CommonsenseQA, COPA, and ATOMIC. The only exception is HardPCR. This is mainly because the inference needed for solving HardPCR is more complex than the other tasks, where we not only need to find the relevant knowledge but also need to replace the target pronouns with the entity in the provided knowledge. As shown in Figure 5, two paths can be found relevant to question: (1) "I am drunk"→$Co\_Occurrence$→"I hit someone"; (2) "I am drunk"→$Co\_Occurrence$→"That is not fair"→$Co\_Occurrence$→"You kick me". For the correct inference, we need to know when there is a conflict, we should trust the one-hop inference more because the additional node in the two-hop path may introduce extra noise. As a comparison, for other tasks, the main inference we need is to find the relevant paths, which is relatively easy. How to train a model that can learn to conduct such complex reasoning is a problem worth exploring in the future.

In general, the observed generalization ability is encouraging because if we can learn a good model on **CIKQA**, based on the assumption that there are limited types of inference, we can potentially solve any commonsense reasoning task as long as the needed inference types are covered by **CIKQA**. At the same time, we also notice that models typically generate better when gold knowledge is provided, further proving the importance
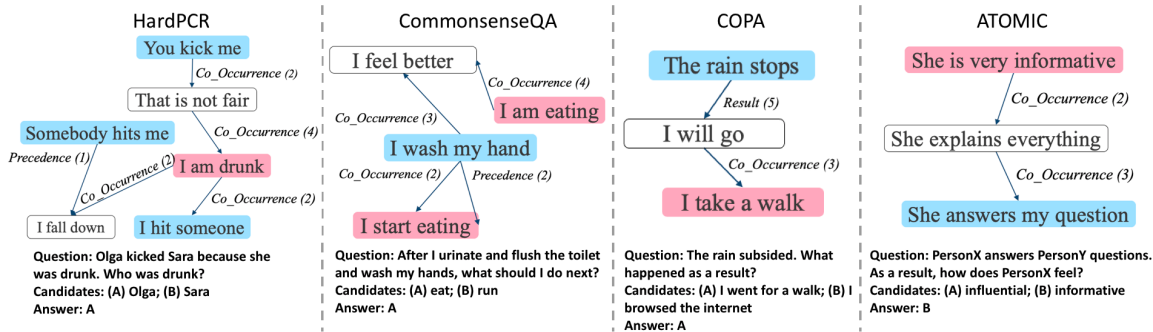
Figure 5: **CIKQA** Case Study. Mapped nodes for the question/answers are in blue/pink. Other nodes are white. Edge weights are in brackets. We only show the relevant parts of the graphs for clear representation.

of the gold knowledge identification task.

5 Related Work

To help machines understand commonsense, the community has devoted great efforts to constructing commonsense knowledge bases with either crowdsourcing (e.g., ConceptNet (Liu and Singh, 2004) and ATOMIC (Sap et al., 2019)) or information extraction techniques (e.g., ASER (Zhang et al., 2020a)). Typically, crowd-sourced knowledge bases are of higher quality, and the auto-constructed ones have broader coverage. Besides acquiring commonsense knowledge, the community also developed many commonsense reasoning datasets to train and test models' commonsense reasoning abilities. Even though these datasets may have different *formats* (e.g., slot fitting in Winogrande (Sakaguchi et al., 2020) and question answering in CommonsenseQA (Talmor et al., 2019)), *knowledge types* (e.g., causal commonsense in COPA (Roemmele et al., 2011) and numerical commonsense in NumerSense (Lin et al., 2020)), or *modalities* (e.g., visual commonsense in VCR (Zellers et al., 2019) and textual commonsense in many others), they follow a standard supervised learning setting, and aim at helping machines to solve a specific commonsense task in an end-to-end manner. Given this setting, it is often difficult to tell what has been learned during the training. Was it used to acquire commonsense knowledge, learn to conduct commonsense inference, or both? Such ambiguity limits our progress in solving these commonsense reasoning tasks. In this work, we connect the efforts on commonsense acquisition and inference by creating a commonsense inference benchmark **CIKQA** , where models can focus on learning to identify the gold knowledge and perform inference over the sup-

porting commonsense knowledge.

Answering questions in natural language based on a knowledge base (KB) is a mature research topic in the NLP community, which is also known as the KBQA problem (Clark et al., 1999; Yih et al., 2015, 2016; Usbeck et al., 2017; Cui et al., 2017). Previous work mainly focuses on factual knowledge, which is stored in the triplets format. The main challenge is to parse the question and then precisely and effectively identify the correct path over a large-scale KB to make the inference. Compared with inference over factual knowledge, inference over commonsense knowledge brings the following unique challenges: (1) Commonsense is a kind of preference rather than fixed knowledge. As a result, the ideal commonsense reasoning process could involve the comparison of multiple candidates. For example, both "drink coffee" and "drink bear" could happen in the morning, but a normal person will prefer "drink coffee;" (2) Beyond named entities, commonsense knowledge also covers daily entities and events, and thus it is difficult to find an exact node from the commonsense KB that matches the question, and we may need to conduct inference based on the partial match (i.e., the extracted nodes are relevant but not identical).

6 Conclusion

In this paper, we present **CIKQA**, a unified commonsense inference benchmark. Specifically, we first convert several popular commonsense tasks into a unified QA format and then equip each question with a supporting commonsense knowledge graph. We also leverage humans to annotate the quality of auto-extracted knowledge. Experiments show that even though models can better learn how to perform commonsense inference with a few ex-

amples and significantly outperform the baseline method that does not use structured knowledge in the data-scarce setting, identifying the gold knowledge is still a challenging problem. More interestingly, with our unified formulation, models demonstrate the encouraging generalization ability across tasks. As both the format unification and supporting graph extraction are automatic, we can easily extend to other commonsense reasoning tasks in the future.

Acknowledgements

Limitations

A common limitation of existing semi-parametric models is the coverage of knowledge resources. **CIKQA** also faces this limitation. Based on our analysis, the largest commonsense knowledge bases can still cover part of the questions in existing commonsense benchmarks. How to populate these commonsense knowledge graphs is an important research question in the future.

References

Ning Bian, Xianpei Han, Bo Chen, and Le Sun. 2021. Benchmarking knowledge-enhanced commonsense question answering via knowledge-to-text transformation. In *Proceedings of AAAI 2021*, pages 12574–12582. AAAI Press.

Peter Clark, John Thompson, and Bruce Porter. 1999. A knowledge-based approach to question-answering. In *Proceedings of AAAI 1999*, pages 43–51.

Amir D. N. Cohen, Shachar Rosenman, and Yoav Goldberg. 2020. Relation extraction as two-way span-prediction. *CoRR*, abs/2010.04829.

Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. 2017. KBQA: learning question answering over QA corpora and knowledge bases. *Proceedings of VLDB 2017*, 10(5):565–576.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL 2019*, pages 4171–4186.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of EMNLP 2020*, pages 671–683.

Yanai Elazar, Hongming Zhang, Yoav Goldberg, and Dan Roth. 2021. Back to square one: Artifact detection, training and commonsense disentanglement in the winograd schema. In *Proceedings of EMNLP 2021*, pages 10486–10500. Association for Computational Linguistics.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of EMNLP 2021*, pages 6894–6910. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *CoRR*, abs/1702.08734.

Jerrold Katz and Jerry Fodor. 1963. The structure of a semantic theory. *Language*, 39:170–210.

Mayank Kejriwal and Ke Shen. 2020. Do fine-tuned commonsense language models really generalize? *CoRR*, abs/2011.09159.

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 *Proceedings of EMNLP 2020 Findings*, pages 1896–1907.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Proceedings of KR 2012*.

Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. 2019. Kagnet: Knowledge-aware graph networks for commonsense reasoning. In *Proceedings of EMNLP-IJCNLP 2019*, pages 2829–2839.

Bill Yuchen Lin, Seyeon Lee, Rahul Khanna, and Xiang Ren. 2020. Birds have four legs?! numersense: Probing numerical commonsense knowledge of pretrained language models. In *Proceedings of EMNLP 2020*, pages 6862–6868.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Hugo Liu and Push Singh. 2004. Conceptnet: a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

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 AAAI 2020*, pages 8449–8456.

Nasrin Mostafazadeh, Aditya Kalyanpur, Lori Moon, David Buchanan, Lauren Berkowitz, Or Biran, and Jennifer Chu-Carroll. 2020. GLUCOSE: GeneraLized and COntextualized story explanations. In *Proceedings of EMNLP 2020*, pages 4569–4586.

Debjit Paul and Anette Frank. 2020. Social commonsense reasoning with multi-head knowledge attention. In *Proceedings of the EMNLP 2020, Findings*, pages 2969–2980.

Altaf Rahman and Vincent Ng. 2012. Resolving complex cases of definite pronouns: The winograd schema challenge. In *Proceedings of CoNLL 2012*, pages 777–789.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Proceedings of AAAI 2011 Spring Symposium*, pages 90–95.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of AAAI 2020*, pages 8732–8740.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. 2022. Multitask prompted training enables zero-shot task generalization. In *Proceedings of ICLR 2022*.

Maarten Sap, Ronan Le Bras, Emily Allaway, Chandra Bhagavatula, Nicholas Lourie, Hannah Rashkin, Brendan Roof, Noah A Smith, and Yejin Choi. 2019. ATOMIC: an atlas of machine commonsense for if-then reasoning. In *Proceedings of AAAI 2019*, pages 3027–3035.

Thibault Sellam, Dipanjan Das, and Ankur P. Parikh. 2020. BLEURT: learning robust metrics for text generation. In *Proceedings of ACL 2020*, pages 7881–7892.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of NAACL 2019*, pages 4149–4158.

Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 2017. 7th open challenge on question answering over linked data (QALD-7). In *Proceedings of 4th SemWebEval Challenge at ESWC 2017*, pages 59–69.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2020. Corefqa: Coreference resolution as query-based span prediction. In *Proceedings of ACL 2020*, pages 6953–6963.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of ACL 2015*, pages 1321–1331.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of ACL 2016*, pages 201–206.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of CVPR 2019*, pages 6720–6731.

Hongming Zhang, Xin Liu, Haojie Pan, Yangqiu Song, and Cane Wing-Ki Leung. 2020a. ASER: A large-scale eventuality knowledge graph. In *Proceedings of WWW 2020*, pages 201–211.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2020b. Winowhy: A deep diagnosis of essential commonsense knowledge for answering winograd schema challenge. In *Proceedings of ACL 2020*, pages 5736–5745.

Hongming Zhang, Xinran Zhao, and Yangqiu Song. 2021. A brief survey and comparative study of recent development of pronoun coreference resolution in English. In *Proceedings of CRAC@EMNLP 2021*, pages 1–11.