Knowledge-Based Systems 235 (2022) 107612

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

PathReasoner: Explainable reasoning paths for commonsense question answering

Xunlin Zhan [a,1], Yinya Huang [a,1], Xiao Dong [b], Qingxing Cao [a], Xiaodan Liang [a,*]

[a] School of Intelligent Systems Engineering, Sun Yat-Sen University, Guangzhou, 510000, China
[b] School of Artificial Intelligence, Sun Yat-Sen University, Zhuhai, 528478, China

ARTICLE INFO

Article history:
Received 16 May 2021
Received in revised form 13 October 2021
Accepted 16 October 2021
Available online 29 October 2021

Keywords:
Commonsense reasoning
Knowledge graph
Reasoning path
Interpretability

ABSTRACT

Commonsense question answering has attracted increasing attention as a challenging task requiring the human reasoning process of answering questions with the help of abundant commonsense knowledge. Existing methods mostly resort to large pre-trained language models and face many difficulties when dealing with the out-of-scope reasoning target, and are unaware of explainable structured information. In this paper, we explore explicitly incorporate external reasoning paths with structured information to explain and facilitate commonsense QA. For this purpose, we propose a PathReasoner to both extract and learn from such structured information. The proposed PathReasoner consists of two main components, a path finder and a hierarchical path learner. To answer a commonsense question, the path finder first retrieves explainable reasoning paths from a large-scale knowledge graph, then the path learner encodes the paths with hierarchical encoders and uses the path features to predict the answers. The experiments on two typical commonsense QA datasets demonstrate the effectiveness of the PathReasoner. The case study gives insightful findings that the reasoning paths provide explainable information for the question answering through the PathReasoner.

1. Introduction

The task of question answering (QA) is widely explored for testing the natural language comprehension as well as the reasoning capability of intelligent systems. For instance, the knowledge-based question answering (KBQA) [1–3] requires the QA systems to read a simple text then reason over a knowledge graph for answering a question. The open-domain question answering [4–6] needs the QA systems to retrieve open-domain documents according to the question and to answer the question with the retrieved information. The multi-hop question answering [7,8] requires the QA systems to reason over multiple documents for answering the question.

Recently, commonsense question answering is proposed [9–12] with challenging commonsense reasoning test while answering the questions. The commonsense reasoning requires QA systems to simulate the human-like capability that makes associations and reasons between the questions and the comprehensive world knowledge. However, different from existing reasoning QA that the knowledge bases are provided by the datasets and highly consistent with the questions, in commonsense QA, the key knowledge information is not provided, including the referred commonsense knowledge and the logic of answering the questions.

The main challenges of commonsense QA are two folds. First, there is no ground truth commonsense knowledge for the questions. The QA systems have to resort to external knowledge bases to find evidence for the questions. The sources and formats of the external evidence are diverse. For example, the evidence can be retrieved from news, Wikipedia, or large-scale knowledge graphs, whereas the formats include structured such as triples or unstructured as the articles. Besides, the evidence can whether be retrieved from existing sources with rules [14] or generated with pre-trained generators conditioned on the questions [15]. The quality of the external evidence, such as the consistency with the questions, restricts the performance of QA systems. Second, the reasoning from the questions to the answers is mostly not intuitive. The QA systems are required to reason over questions via multiple steps to obtain the correct answers. An example is illustrated in Fig. 1. The question is about what "*children*" need to "*grow up*" "*healthy*". The five answer options are "*watch television*", "*wash dishes*", "*come home*", "*need care*" and "*fast food* ". From the figure, the extracted subgraphs show that the "*children*" entity is directly related to "*wash dishes*, "*watch television*", "*come home*" and "*need care*". "*children*" is indirectly related to fast food through the "*Children –candy–junk food–fast food*" pathway.

* Corresponding author.
E-mail addresses: zhanxlin@mail2.sysu.edu.cn (X. Zhan), yinya.el.huang@gmail.com (Y. Huang), dx.icandoit@gmail.com (X. Dong), caoqx@mail2.sysu.edu.cn (Q. Cao), xdliang328@gmail.com (X. Liang).

[1] Equal contribution.

https://doi.org/10.1016/j.knosys.2021.107612
0950-7051/

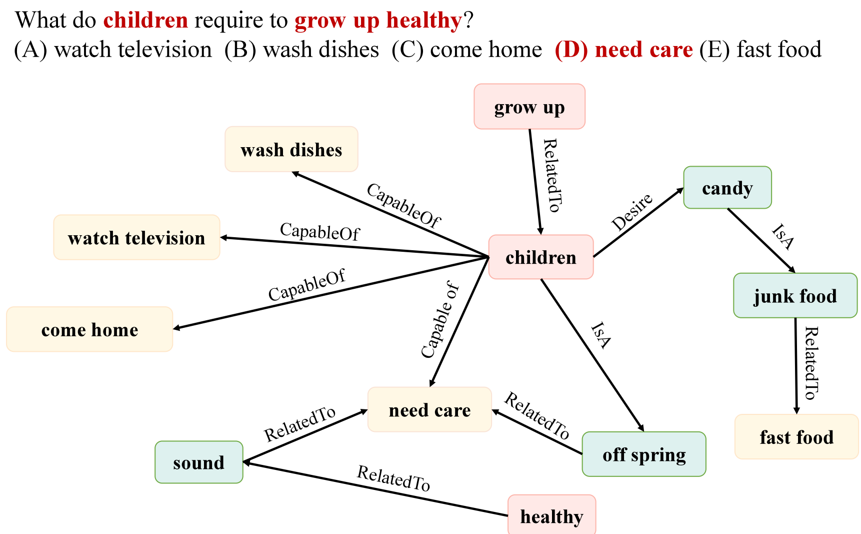


Fig. 1. The main challenge of the Commonsense QA task is that the relations between the question and the answer options are not intuitive, and therefore multi-hop reasoning processes are needed. The question is taken from the CommonsenseQA dataset [9] and the entity graph is a sub-graph of the ConceptNet [13].

However, for node "*need care*", the pathway "*healthy–sound–need care*" provides background knowledge on the need for care to stay healthy, and the pathway "*children– off spring–need care*" provides background knowledge about the need for care offspring. While such background knowledge is common knowledge to human beings, it is of great significance for machines.

To obtain the commonsense evidence, Rajani et al. [15] manually annotate explanations for each commonsense question. Lin et al. [14] construct schema graphs based on the questions. Lv et al. [16] extract knowledge from multiple sources including ConceptNet [13] and Wikipedia. Asai et al. [17] introduce a new graph-based recurrent retrieval approach that learns to retrieve reasoning paths over the Wikipedia graph to answer multi-hop open-domain questions. [18] generates context-dependent clauses, which form a dynamic Knowledge Graph (KG) on-the-fly for commonsense reasoning. As for the models, existing methods are mainly characterized into two types: (1) rule-based methods, (2) deep learning models. The rule-based methods are intuitive such as choosing the answer with top cosine similarity to the question. Such methods act as baselines, and most of them do not involve external commonsense knowledge. Deep learning models incorporate commonsense knowledge implicitly or explicitly. The implicit approaches [9,10] incorporate the knowledge by pre-training, such as masking the entities in the evidence [19] then using commonsense-related supervision [20]. The explicit methods encode the commonsense knowledge with memory networks [21] or graph models [14,16,22]. Li et al. [23] explore three token-level injection methods to extend BERT to allow flexible incorporation of external knowledge and introduce a masking mechanism for a token-level multi-hop relational search to filter external knowledge. Due to the lack of capability to capture symbolic logic, Wang et al. [24] propose to understand logical symbols and expressions in the text to arrive at the answer. However, the external commonsense knowledge from current methods is not explainable enough since the reasoning from the question to the correct answer is not intuitive and mostly requires multi-hop reasoning. Therefore, structured information should be involved.

In this paper, we propose a PathReasoner that extracts and learns explainable commonsense knowledge over question answering. It consists of two main components, a path finder and a hierarchical path learner. The path finder extracts the explainable paths based on the commonsense questions. The hierarchical path learner then encodes the multi-hop entity paths, conducts soft selection among the paths, and predicts the answer.

The main contributions of this paper can be summarized as follows:

- We propose a reasonable and explainable framework to explicitly incorporate external reasoning paths with structured information to explain and facilitate commonsense QA.
- We use a Hierarchical Path Learner including an intra-path encoder and an inter-path encoder to reason the path over questions.
- The experiments on two typical commonsense QA datasets demonstrate the effectiveness of our proposed PathReasoner.

The rest of the paper is organized as follows. In Section 2, we briefly review the related works. In Section 3, we introduce our method. We first give an overview in Section 3.1, then introduce the path finder module in Section 3.2 and the path learner module in Section 3.3. After that introduce the answer prediction module in Section 3.4. Experimental results and discussions are addressed in Section 4. Section 5 concludes the paper and describes the future works.

2. Related works

2.1. Question answering with reasoning

A large number of question answering datasets test QA systems' multiple reasoning capabilities. WSC [25] ask single binary questions that contain two proper nouns and one pronoun. The QA systems are required to recognize one of the two proper nouns that the pronoun is referring to according to the sentence. bAbI [3] measures QA systems via chaining facts and simple reasoning. For this purpose, each sample contains one question and three to four sentences as background. The answers are mostly the entities in the background sentences or yes/no selections. The reasoning types for the answering include counting, negation, size reasoning, time reasoning, and so forth. Free917 [2] and WebQuestion [1] provide questions and their logical forms. QA systems need to answer the questions by reasoning over the Freebase[2] entities.

2 www.freebase.com.

Besides, the open-domain QA such as MCTest [4], CNN/Daily Mail [5] and NewsQA [6] provide multiple documents for each of the questions, and the evidence for answering the questions needs to be selected by the QA systems from one of the documents. The multi-hop QA such as HotpotQA [7] and WikiHop [8] need the QA systems to reason over multiple documents which associate the questions with the answers. Reasoning with such abundant information is more challenging.

Different from previous QA tasks, the recently proposed commonsense QA datasets need external commonsense knowledge which is not provided by the questions [9,10,12]. Besides, the reasoning from the questions to the answers is not intuitive, and therefore answering the questions needs multi-hop reasoning over the unlabeled commonsense knowledge. CommonsenseQA [9] ask questions with commonsense in daily life, and the answer options are mostly entities seen in daily life. WIQA [10] contains more challenging "what-if" questions about scientific facts. It provides a short paragraph for each question describing some natural phenomenon, but such paragraphs mostly do not cover the information needed for answering the questions. Therefore external commonsense knowledge is still needed for the answering. From this point of view, commonsense QA is more challenging than previous QA datasets. And the existing QA systems are limited in solving this task.

2.2. QA systems for reasoning

QA systems conduct the reasoning with multiple approaches, for example, memory networks and graph models. The first group of QA systems saves and selects the knowledge with memory networks [26,27]. DMN [28] uses a dynamic memory network to form episodic memories then generate relevant answers. KV-MemNN [29] defines the memory slots as key–value pairs and it answers the questions by querying the pairs with the questions. REM-Net [30] use a memory network with an erasure operation to select the useful information from the referred knowledge. [31] integrates multiple knowledge sources such as ConceptNet, Wikipedia, and the Cambridge Dictionary and proposes an answer choice-aware attention mechanism to fuse all hidden representations.

The second group of systems uses graph models. [32] incorporates the question with knowledge subgraphs or paths that carry information such as relation among concepts or show multi-hop reasoning process. [33] introduces a heterogeneous graph with different granularity levels of information including candidates, documents and entities in specific document contexts and employs Graph Neural Networks (GNN) based message passing algorithms to accumulate evidence on the heterogeneous graph. [16] extracts both structured knowledge which is from the ConceptNet and unstructured knowledge from Wikipedia and converts the Wikipedia knowledge into structured. The knowledge features are encoded and updated with a graph convolutional network. [14] builds semantic graphs of the questions based on the large-scale knowledge graph, and the graph features are learned by a graph convolutional network following by an LSTM [34]. [35] proposes an AMR-ConceptNet-Pruned (ACP) graph pruned from a full integrated graph to interpret the reasoning path and predict the correct answer. In multi-domain, the GRUC [36] depicts the image with multiple graphs in multiple modalities, including semantic graph, visual graph, and fact graph. It then encodes the graphs with graph convolution, after which selects and integrates the graph knowledge with a memory-based module.

2.3. Large-scale knowledge graphs

Several large-scale knowledge graphs are built for demonstrating the word semantics or entity relations. Such graphs provide supplementary knowledge for the reasoning. For example, WordNet [37] and ConceptNet [13] are semantic networks that connecting words with semantic relations. The relations in WordNet are mostly lexical relations such as hyperonymy, hyponymy, or "IsA" relations. The relations in ConceptNet are more comprehensive, including causal relationship ("Causes"), purpose ("UsedFor"), location ("AtLocation") and so forth. The words/phrases in such semantic networks mostly denote entities or their descriptions, therefore the relations describe commonsense knowledge in the world. Besides, there are knowledge graphs representing relations between events rather than entities. Such knowledge graphs include ATOMIC [38] and ASER [39]. The graph nodes in ATOMIC or ASER are events such as "X repels Y's attack", "I have lunch". The relations between the events are cause–effect, reason, result, and so forth. Such eventuality knowledge graphs also carry commonsense knowledge. In this paper, the commonsense knowledge we need for our solution is mostly entity-based. Therefore we extract the knowledge from the large-scale ConceptNet.

3. PathReasoner

The proposed PathReasoner solves commonsense QA by reasoning over explicit explainable reasoning paths that simulate the process of human reasoning from the question to the answer options. The overall architecture of the PathReasoner is demonstrated in Fig. 2. It is composed of two main modules, a path finder and a hierarchical path learner. The path finder retrieves explainable reasoning paths from a large-scale knowledge graph, whereas the hierarchical path learner encodes the retrieved reasoning paths, learns the path features for commonsense answer prediction.

3.1. Overview

Given a commonsense question with a question sentence along with several answer options, the PathReasoner conducts the reasoning then gives the probabilities of each answer option so as to choose the correct answer to the question. The reasoning process includes finding paths from the question to each answer option given the knowledge graphs, after which learning the path features over the questions. Building the explicit reasoning paths resorts to a large-scale knowledge graph. The PathReasoner first extracts key entities from the question sentences, then align the key question entities and the answer options with the entities in the knowledge graph. When finding reasoning paths, the path finder searches for neighbor entities of the question entities as well as the answer entities by the entity relations in the knowledge graph. The multi-hop searching returns end-to-end entity triples as paths from the question entities to the answer entities. For learning the retrieved reasoning paths, the hierarchical path learner groups the paths by their shared endpoint answer options. The paths in the same group together are first encoded by an intra-path encoder, then encoded by an inter-path encoder, after which the path features are merged into a feature representing the current answer option based on the question and the reasoning paths. The features for all answer options are fed into a classifier, which returns the probabilities for the answer prediction.

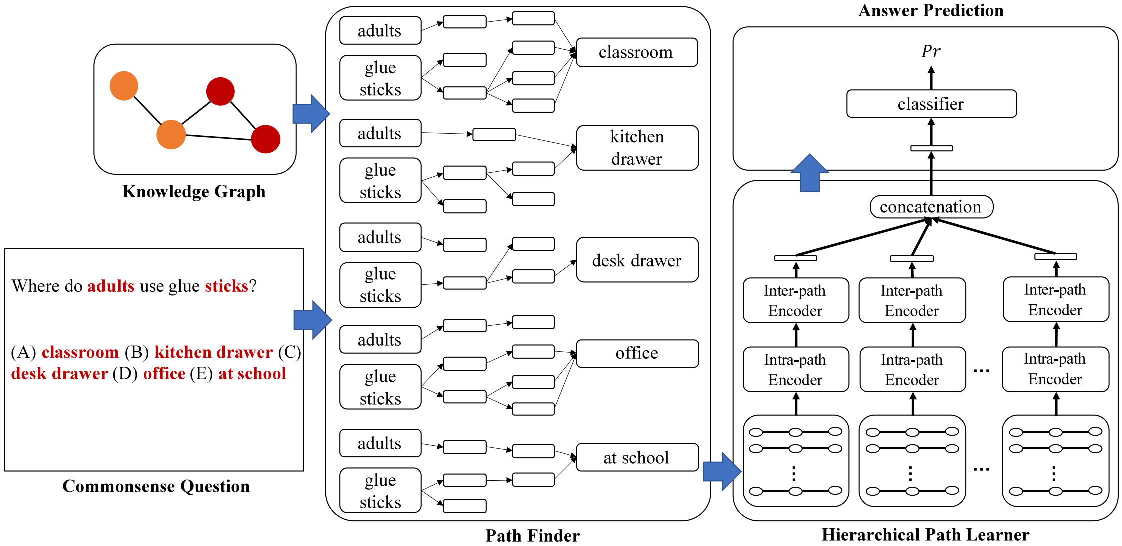


Fig. 2. The proposed PathReasoner consists of two main components: (1) the path finder retrieves reasoning paths for the commonsense question, (2) the path learner hierarchically encodes the reasoning paths for commonsense reasoning.

3.2. Path finder

In this paper, the reasoning paths are directed from the question to each answer option, bridging the question and the answer options with explicit reasoning processes. The reasoning paths resort to large-scale knowledge graphs built with abundant entity relations so that they are essentially end-to-end entity knowledge triples. The starting entity in a path is always from the question, whereas the ending entity is always from one of the answer options.

To obtain such reasoning paths, we design the path finder as demonstrated in Fig. 3. Given a question and its answer options, the path finder first extracts key entities from the question sentence. Then the extracted question entities and the answer options are aligned with the entities in the knowledge graph. With the aligned question entities and answer options, the path finder then conducts a breadth-first search in the knowledge graph from both the question end and the option ends, until both searching paths meet and form complete reasoning paths.

For the key-entity extraction and alignment, the path finder first conducts lemmatization to the question sentence, so that each token in the question is reduced to its base form. For example, the question in Fig. 3 "*Where do adults use glue sticks?*" is reduced to "*Where do adult use glue stick?*". Then the n-grams in the reduced question sequence are compared to the knowledge graph entities. The matched n-grams are saved as question key entities. Similarly, the answer options are first lemmatized, then matched to the knowledge graph entities.

The question entities are taken as starting points of the reasoning paths, whereas the answer options are taken as endpoints. The breadth-first search is conducted from both ends simultaneously. To control the time and space complexity, in the meantime to ensure the quality of reasoning paths without unnecessary detours, we first set the maximum hop H of the retrieved reasoning paths and set the maximum neighbors N in each searching step. The path finder searches $\lceil H/2 \rceil$ steps from both the question entities and the answer entities. When two sub-paths share the last node, they are concatenated after popping out one of the last nodes. The complexity of this bidirectional breadth-first path finding is $\mathcal{O}(N^{\lceil H/2 \rceil})$.

For a preliminary estimation of the reasoning path quality, the path finder evaluates the correlation of each reasoning path and the question sentence with BERTScore [40]. The reasoning paths are then sorted with the BERTScore. The paths with low BERTScores are filtered out with a threshold.

3.3. Hierarchical path learner

Given the retrieved reasoning paths, the objective of the Path Learner is to estimate and select the paths with the path features. Such learned path features are used for downstream multi-choice answer prediction. The architecture of the Path Learner is demonstrated in Fig. 4. The path features are encoded and learned with hierarchical encoders, including an intra-path encoder and an inter-path encoder. The input of the Hierarchical Path Learner is a group of reasoning paths ending in the same answer option, whereas the output of the Hierarchical Path Learner is the corresponding path features for each input reasoning path.

3.3.1. Intra-path encoder

The intra-path encoder is to encode each reasoning path along with the question sentence, so that to obtain the path features in the context of the question. To do this, the reasoning paths are first regarded as token sequences whereas the end-to-end entity triples are flattened to entity and relation sequences. For example, the reasoning path "`(revolving door, at_location, mall), (mall, at_location, New York)`" is flattened to "`revolving door at location mall at location New York`". The question sentence and the flattened path sequence are then concatenated into one sequence delimited by special tokens (e.g., `[CLS]` and `[SEP]` for BERT [41]). For example, the question in Fig. 1 and the above-mentioned flattened path are concatenated as "`[CLS] A revolving door is convenient for ... at a what ? [SEP] revolving door at location mall at location New York [SEP]`". All the concatenated question-path pairs are fed into a shared encoder E_{intra}. In practice, E_{intra} is a pre-trained language model such as BERT [41] or RoBERTa [42], and the outputting first token embedding (e.g., `[CLS]` for BERT) is taken as the contextual intra-path embedding:

$$\mathbf{e}^{(k)} = \mathbf{e}_0^{(k)} = E_{intra}([t_0^{(k)}; t_1^{(k)}; \dots; t_L^{(k)}]), \tag{1}$$

where $\mathbf{e}^{(k)} \in \mathbb{R}^{d_{model}}$ and d_{model} is the embedding size. $t_*^{(k)}$ are the tokens in the input sequence, L is the sequence length, $k \in K$ and K is the number of paths with shared endpoints.

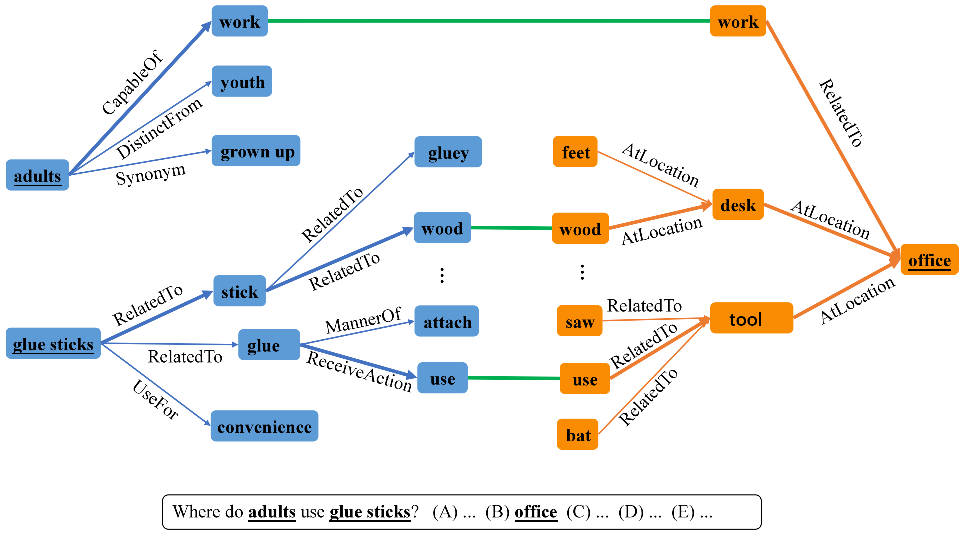


Fig. 3. The bidirectional breadth-first path finding from the question to one of the answer options. The search processes are simultaneously conducted from the question end and the answer end. Once the retrieved sub-paths from both ends meet up with a shared entity node (green line), they are joined into complete reasoning paths.

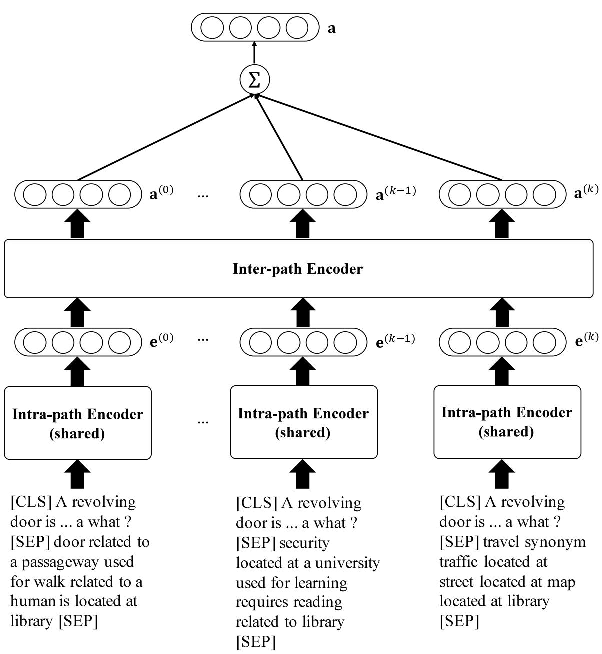


Fig. 4. The hierarchical path learner with an intra-path encoder and an inter-path encoder. The inputs are the reasoning paths that end in the same answer option, which are concatenated with the question sentence with special tokens (e.g., `[CLS]`, `[SEP]`). The outputs are path features for each path, which are then merged by summation. The intra-path encoders encoding each path share the weights.

3.3.2. Inter-path encoder

While the intra-path encoder focuses on single path features with the question context and encodes the paths independently, the following inter-path encoder conducts interactions among the set of paths, which obtain higher-level path features and conduct a soft selection with weights calculated by softmax among the paths. The more relevant the question path, the larger the weight. For example, the path "glue sticks - stick -wood - desk - office" whose weight is small is irrelevant in Fig. 3.

Given the intra-path features $\mathbf{E} \in \mathbb{R}^{K \times d_{model}}$, where $\mathbf{E} = [(\mathbf{e}^{(0)})^\top; (\mathbf{e}^{(1)})^\top; \ldots; (\mathbf{e}^{(K)})^\top]$ and K is the number of paths with shared endpoints, the inter-path encoder conducts multi-head inter-path self-attention over the path features:

$$\mathrm{InterAtt}_h(\mathbf{E}) = \mathrm{softmax}\left(\frac{(\mathbf{E}^\top W_h^Q)(\mathbf{E}W_h^K)}{\sqrt{d_{model}}}\right)\mathbf{E}W_h^V, \tag{2}$$

$$\mathrm{MultiHead}(\mathbf{E}) = \mathrm{Concat}(\mathrm{InterAtt}_1, \ldots, \mathrm{InterAtt}_H)W^O, \tag{3}$$

where H is the number of heads and $h \in \{1, 2, \ldots, H\}$, W_h^Q, W_h^K, W_h^V, $W^O \in \mathbb{R}^{d_{model} \times d_{model}}$.

Therefore the multi-head inter-path self-attention outputs the inter-path features:

$$[(\mathbf{a}^{(0)})^\top; (\mathbf{a}^{(1)})^\top; \ldots; (\mathbf{a}^{(K)})^\top] = \mathrm{MultiHead}(\mathbf{E}), \tag{4}$$

where $\mathbf{a}^{(*)} \in \mathbb{R}^{d_{model}}$ are the inter-path features for each path and K is the number of paths.

3.3.3. Multiple-choice path features

The intra-path encoder and the inter-path encoder are applied to the groups of paths with shared end-point answer options separately. We denote $\mathbf{A}_c = [(\mathbf{a}_c^{(0)})^\top; (\mathbf{a}_c^{(1)})^\top; \ldots; (\mathbf{a}_c^{(K)})^\top]$ as the cth group of path features. Each group of path features are merged into a single output feature as:

$$\mathbf{a}_c = \sum_k^K (\mathbf{a}_c^{(k)}), \tag{5}$$

where $\mathbf{a}_c \in \mathbb{R}^{d_{model}}$. Then with C multiple choices, the hierarchical encoders output $\{\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_C\}$ where C is the number of answer options.

3.4. Answer prediction

With the path features for each answer option, the multiple-choice answer prediction is conducted with a linear classification and return the probabilities Pr of choosing the answer options:

$$Pr = \mathrm{softmax}([\mathbf{a}_0; \mathbf{a}_1, \ldots; \mathbf{a}_C]\mathbf{W} + \mathbf{b}), \tag{6}$$

where [;] indicates feature concatenation, $\mathbf{W} \in \mathbb{R}^{d_{model} \times C}$ and $\mathbf{b} \in \mathbb{R}^C$ are weights and bias of the classifier.

Table 1
Experimental results compared with multiple groups of methods on the WIQA dataset. "In", "out" and "no" indicates the "in-para" questions, "out-of-para" questions, "no-effect" questions in WIQA test set respectively. "Total" indicates the results on the overall test set (%).

Method	In	Out	No	Total
Rule-based Models				
Majority [10]*	45.46	49.47	0.55	30.66
Polarity [10]*	76.31	53.59	0.27	39.43
Adaboost [43]*	49.41	36.61	48.42	43.93
Deep Models				
Decomp-Att [10]*	56.31	48.56	73.42	59.48
BERT-Base	70.57	58.54	91.08	74.26
BERT-Large	73.40	63.88	90.52	76.69
RoBERTa-Base	73.58	61.41	**92.27**	76.64
RoBERTa-Large	74.91	67.08	90.20	78.12
Explicit Reasoning Paths				
MemN2N [21] + paths	38.50	38.01	39.52	38.85
BERT-Base + paths	70.57	61.00	90.72	75.12
BERT-Large + paths	73.40	63.88	90.52	76.69
RoBERTa-Base + paths	75.85	64.94	89.80	77.26
RoBERTa-Large + paths	76.98	68.88	90.44	79.32
Ours				
PathReasoner (BERT-Base)	73.02	61.66	91.71	76.22
PathReasoner (BERT-Large)	74.91	66.17	91.79	78.42
PathReasoner (RoBERTa-Base)	77.55	70.03	89.96	79.55
PathReasoner (RoBERTa-Large)	**77.92**	**70.69**	91.55	**80.69**

Table 2
Experimental results compared with multiple groups of methods on the CommonsenseQA dataset (%).

Methods	Dev Acc
Deep Models	
RoBERTa-Base [41]	65.36
RoBERTa-Large [42]	76.24
Explicit Reasoning Paths	
MemN2N [21] + paths	26.78
BERT-Base + paths	57.82
BERT-Large + paths	64.37
RoBERTa-Base + paths	66.26
RoBERTa-Large + paths	76.58
Ours	
PathReasoner (BERT-Base)	59.38
PathReasoner (BERT-Large)	64.54
PathReasoner (RoBERTa-Base)	68.46
PathReasoner (RoBERTa-Large)	**77.81**

4. Experiments

4.1. Datasets and evaluation metrics

CommonsenseQA The CommonsenseQA dataset [9] is built upon the ConceptNet [13]. It contains 12,247 questions where each question is a single sentence along with five answer options. The answer options are entities in the ConceptNet, whereas the question sentences are manually built with other related entities. The questions are randomly split into 9,741/1,221/1,140 training/dev/test data.

WIQA The WIQA dataset [10] also contains multiple-choice questions with three answer options. The questions are about commonsense such as natural phenomenon. For each question, a procedural text is provided as a reference to answering the question. The total 40,695 questions is randomly split into 29,808/6,894/3,993 training/dev/test data. The questions within the training/dev/test split may share the procedural text, but questions from different splits do not share. The test set is further divided into three types of questions including the "in-para" questions, the "out-of-para" questions, and the "no-effect" questions according to whether the questions are derived from or can be answered by the given procedural text.

Evaluation metrics Both of the datasets are multiple-choice QA tasks and are evaluated by the accuracy of choosing the correct answer options. Besides, since the WIQA test set is separated into three subsets ("in-para", "out-of-para", "no-effect"), it also evaluates the prediction accuracy for each subset. The prediction accuracy can be calculated via

$$Acc = \frac{Correct\ Prediction\ Sample\ Numbers}{All\ Sample\ Numbers}. \tag{7}$$

4.2. Implementation details

4.2.1. Details of the path finder

Key entities extraction We use an off-the-shelf toolkit, TAGME[3] [44], to extract key entities from the question sentences. The TAGME toolkit is capable of identifying short phrases and links them to Wikipedia pages, therefore the extracted phrases are mostly meaningful entities. There is a parameter controlling the density of identified phrases. In our experiments, we manually tune the parameter for obtaining more reasonable extractions, and we set the parameter to 0.1.

Searching for reasoning paths The large-scale knowledge graph we use for the path searching is ConceptNet [13]. The extracted question key entities are aligned with the entities in ConceptNet as the starting points of the reasoning paths. The alignment is conducted by lemmatization of both the question key entities and the ConceptNet entities, and the entities are aligned once they share a base form. Besides, the answer options are also aligned with the ConceptNet entities in the same manner. When conducting the bidirectional breadth-first search, the maximum hop of reasoning paths is set to 4, and the maximum of neighbors in each search step is set to 200.

4.2.2. Details of the hierarchical path learner

The input sequence length to the path learner is 128. For the intra-path encoder, we fine-tune the BERT [41] and the RoBERTa [42] as two variant experimental settings. We denote the corresponding PathReasoner variants as PathReasoner (BERT) and PathReasoner (RoBERTa) respectively. The input sequences to the PathReasoner variants use the special tokens corresponding to the pre-trained models, which means the input to the PathReasoner (BERT) are "`[CLS]` *path* `[SEP]` *question* ∥ *option* SEP", and the input to the PathReasoner (RoBERTa) are "`<s>` *path* `</s>` *question* ∥ *option* `</s>`", where "∥" denotes sequence concatenation. For the inter-path encoder, the number of heads is 8. The learning rate for training the PathReasoner (BERT) is 1e-5, whereas for training the PathReasoner (RoBERTa) is 5e−6. The training is warmed up in the first 10% steps. The parameters are updated with an Adam [45] optimizer. The models are trained 7 epochs with a batch size of 8.

4.3. Compared methods

We compare our PathReasoner with three groups of methods: (1) rule-based methods, (2) deep models without the reasoning paths, (3) deep models with the reasoning paths.

Rule-based methods Majority [10] takes the most frequent answer option in the training set as the prediction. Polarity [10] calculates the frequencies of comparatives in the questions. Adaboost [43] learns bag-of-words features for classification. VecSim [9] predicts the answers according to the cosine similarity between the question and the answer options based on GloVe [46] or Numberbatch [13] embeddings.

[3] https://tagme.d4science.org/tagme/.

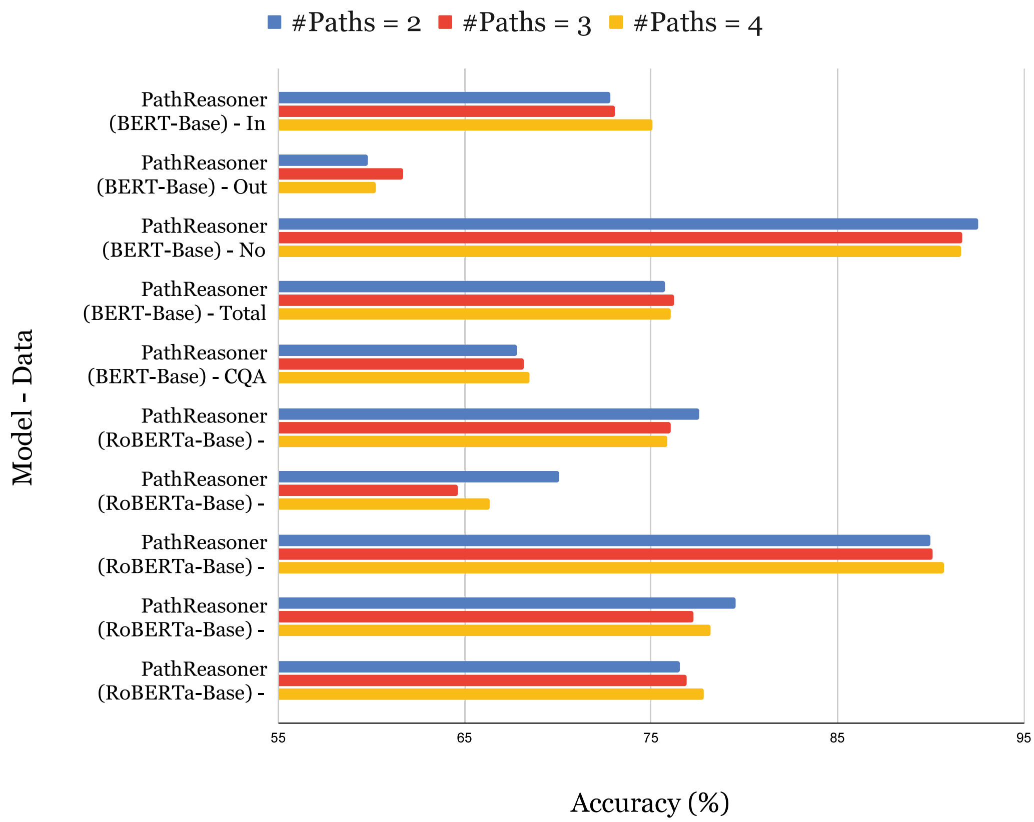


Fig. 5. Ablation study on the number of reasoning paths. "* - In" means the model evaluated on the WIQA "in-para" test questions. "* - Out" means the model evaluated on the WIQA "out-of-para" test questions. "* - No" means the model evaluated on the WIQA "no" test questions. "* - Total" means the model evaluated on the WIQA "total" test questions. "* - CQA" means the model evaluated on the Commonsense dev set.

Deep models without the reasoning paths Decomp-Attn [47] conducts decomposable attention to facilitate sentence-level reasoning. LM1B [9] pre-trains a language model on the One Bilion Words Benchmark [48]. The pre-trained language models such as BERT [41], RoBERTa [42] and GPT [49]) are trained upon abundant corpora and possess a certain amount of commonsense knowledge. But such pre-trained language models do not aware of structured information such as explicit reasoning paths.

Deep models with the reasoning paths The end-to-end memory networks [21] put our retrieved reasoning paths into the memory slots and learn the path features by encoding and selecting the paths based on the commonsense questions with the memory mechanism. The pre-trained language models (BERT [41] and RoBERTa [42]) are fine-tuned by taking the question-path pairs as model inputs. It means that the external reasoning paths are concatenated with the commonsense questions into combined sequences.

4.4. Experimental results

4.4.1. Experimental results on WIQA

The results on WIQA are demonstrated in Table 1. It is shown that adding the external reasoning paths to the deep models results in performance improvement. This indicates the effectiveness of the reasoning paths retrieved by the PathFinder module. For example, RoBERTa-Large with external reasoning paths reaches an accuracy of 79.32%, 1.2% over the pre-trained RoBERTa-Large without the reasoning paths fine-tuning. The end-to-end memory network results with the reasoning paths are relatively poor. This may due to the large difference in the number of parameters between the LSTM-based memory networks and the Transformer-based language models, as well as the pre-training of the language models.

The PathReasoner further improve the performance. The PathReasoner (RoBERTa-Large) reaches a total accuracy of 80.69%, accuracies of "in-para" / "out-of-para" / "no-effect" questions of 77.92% / 70.69% / 91.55%. The "no-effect" accuracy does not outperform the RoBERTa-Base result, and one of the probable reasons is that the "no-effect" questions require less external evidence than the other two questions. The PathReasoner (RoBERTa-Base), PathReasoner (BERT-Large), pathReasoner (BERT-Base) also outperform their corresponding deep models with the external reasoning paths.

4.4.2. Experimental results on CommonsenseQA

The same observations are found in the CommonsenseQA dataset. The experimental results are demonstrated in Table 2. The explicit reasoning paths on the end-to-end memory network are still poor. But the BERT and RoBERTa models with the external reasoning paths outperform the language models. The PathReasoner further improves the performances. Therefore the way the PathReasoner encodes and selects the reasoning paths is effective. Rather than indistinguishably encoding and learning the path features as the compared methods, the PathReasoner learns to softly select the reasoning paths so that the paths being more relevant or with higher quality are allocated with higher weights. This results in a more reasonable use of the paths.

4.5. Ablation study

4.5.1. Ablation study on the hierarchical path learners

We first conduct ablation studies on the components of the PathReasoner. we remove the inter-path encoder from the hierarchical encoders and directly merge the output features from the intra-path encoder to represent the answer option features.

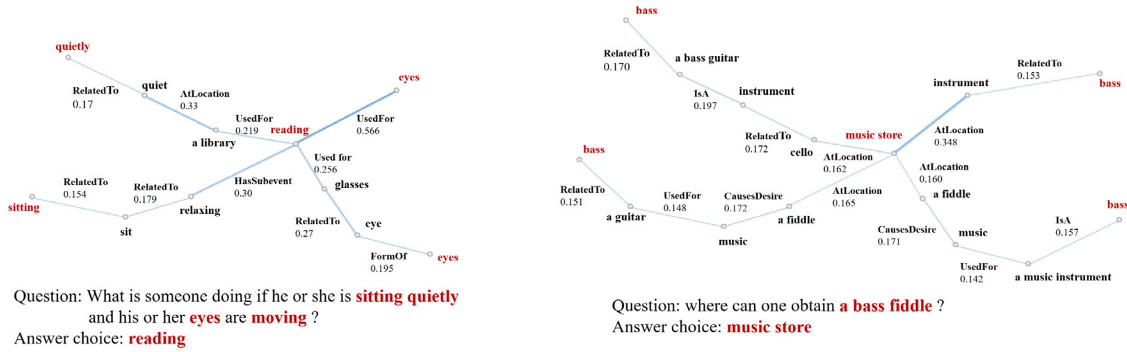


Fig. 6. The behavior of the intra-path encoder. The questions are taken from the CommonsenseQA dataset.

Table 3
Ablation study on the components of the PathReasoner (%).

	WIQA				CommonsenseQA
	In	Out	No	Total	Dev
PathReasoner (BERT-Base)	73.02	61.66	91.71	76.22	59.38
w/o inter-path encoder	73.77	58.37	92.91	75.52	58.15
PathReasoner (RoBERTa-Base)	77.55	70.03	89.96	79.55	68.46
w/o inter-path encoder	73.40	65.93	90.36	77.46	67.73

The results are shown in Table 3. The experiments are conducted on two backbones, i.e., BERT-Base and RoBERTa-Base. The results on both PathReasoner (BERT-Base) and PathReasoner (RoBERTa-Base) demonstrate performance drop without the inter-path encoder. Besides, the same conclusions can be driven from experiments on the WIQA dataset and CommonsenseQA dataset. Therefore the effectiveness of the inter-path encoder is justified.

Further removing the intra-path encoder will bring the model back into the compared language models with the path inputs. The results can be found in Tables 1 and 2. This proves the benefits of the intra-path encoder.

4.5.2. Ablation study on the number of reasoning paths

We then investigate the number of reasoning paths that each time the PathReasoner can learn. During the experiments, we set the number of reasoning paths to 2, 3, 4 respectively, and the experimental results are demonstrated in Fig. 5.

For the CommonsenseQA dataset, the augmentation of the reasoning paths facilitates the PathReasoner better for the answer prediction, since the accuracy grows when the number of reasoning paths increases. For the WIQA dataset, the situation is more complicated. For example, the PathReasoner (BERT-Base) with four reasoning paths performs much better on the "in-para" questions than using two or three paths, but three reasoning paths are sufficient for the "out-of-para" questions. The PathReasoner (RoBERTa-Base) with two reasoning paths performs better in general.

4.6. Case study

In this section, we further explore the capability of the PathReasoner, specifically the behavior of the intra-path encoder and the inter-path encoder. In the hierarchical path learner, the intra-path encoder learns each reasoning path independently, and the inter-path encoder learns the inter-path features by conducting a soft selection among the paths. We visualize the weights learned by both encoders.

4.6.1. Intra-path encoder behavior

We visualize the triple weights learned by the intra-path encoder of the PathReasoner (BERT-base) that is trained on the CommonsenseQA dataset, which is demonstrated in Fig. 6.

The question on the left-hand side contains key entities "*sitting quitely*", "*eyes*" and "*moving*", whereas the answer option is "*reading*". The intra-path encoder provides weights for each triple in the reasoning paths from the question entities to the answer option. Among the triples, the triple (*eyes*, *UsedFor*, *reading*) obtains the highest weight. Since this triple indicates a reasonable relation from "*eyes*" to "*reading*", and provides an explanation to the answer option "*reading*", it is intuitive that it obtains such high weight. Besides, the triple (*quiet*, *AtLocation*, *a library*) in another reasoning path has a weight of 0.33, indicating that this entity relation is significant in this reasoning path. It is this triple that bridges a description of the atmosphere (i.e., "*quiet*") in the question and the place that matches the description in answer option (e.g., "*a library*" for "*reading*").

The question in the right-hand side is about the location of "*a bass fiddle*", and the correct answer option is "*music store*". The explicit reasoning paths indicate the reasoning process from "*a bass fiddle*" to "*music store*". The path "(*bass*, *RelatedTo*, *instrument*), (*instrument*, *AtLocation*, *music store*)" first reduces "*a bass fiddle*" to its category, which is an "*instrument*", then leads to the answer option "*music store*" by the commonsense entity relation that an "*instrument*" usually appears in a "*music store*". The triple weight of "(*instrument*, *AtLocation*, *music store*)" achieves a high 0.348, through which we find that this triple provides key information and help locating the answer option.

4.6.2. Inter-path encoder behavior

We then explore the soft path selection by the inter-path encoder. The attention weights for the reasoning paths are demonstrated in Fig. 7.

The question asks about the causal relationship between "*boiling point*" and "*evaporation*", and the answer option is "*more*" among the three options ("(A) more, (B) less, (C) no effect"). This answer indicates that reaching the "*boiling point*" is positively related to the phenomenon "*evaporation*". The inter-path encoder softly selects relevant paths from four reasoning paths (demonstrated in the caption of Fig. 7). As shown in the figure, the first two paths obtain higher weights, which indicates that these two paths are more relevant to the question.

The question in the right-hand side asks the causal relationship between "*magma*" and "*large mountains*". The correct answer option is "*more*". The four reasoning paths are presented in the caption of Fig. 7. The inter-path encoder selects the last two paths from the four. The entity "*volcano*" in the last two paths provides information to the causal relationship between "*magma*" and "*large mountains*". On the contrary, "*structure*" and "*mass*" in the first two paths are more are less relevant to the causal relationship.

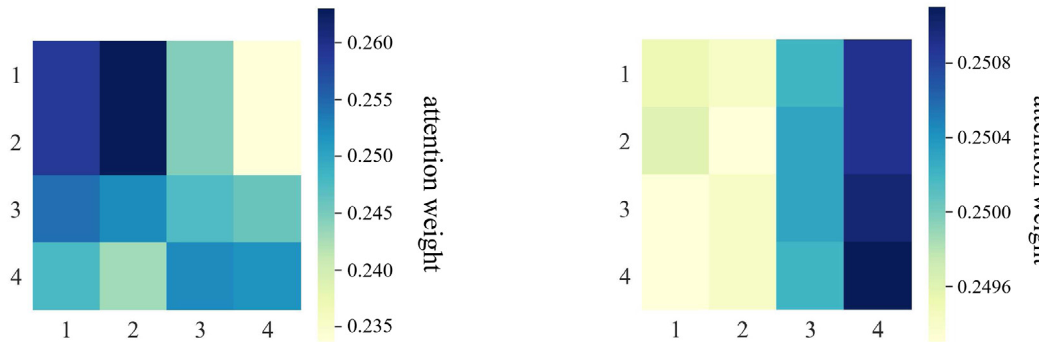


Fig. 7. Behavior of the inter-path encoder. **Left:** Question: "suppose during **boiling point** happens, how will it affect more **evaporation**".. hoices: "more", "less" and "no effect". Correct answer option: "more". The reasoning paths: (1) "(boiling point, RelatedTo, liquid), (liquid, RelatedTo, evaporation)", (2) "(boiling point, RelatedTo, concentration), (concentration, RelatedTo, evaporation)", (3) "(boiling point, RelatedTo, desuperheat), (desuperheat, RelatedTo, vapour), (vapour, RelatedTo, evaporation)", (4) "(boiling point, HasContext, chemistry), (chemistry, HasContext, cryophorus), (cryphorus, RelatedTo, evaporation)". **Right:** Question: "suppose more **magma** is forced up happens, how will it affect **larger mountains**". Correct answer option: "more". The reasoning paths: (1) "(magma, RelatedTo, structure), (structure, RelatedTo, mountain), (mountain, RelatedTo, large)", (2) "(magma, RelatedTo, mass), (mass, RelatedTo, mountain), (mountain, RelatedTo, large)", (3) "(magma, RelatedTo, volcano), (volcano, RelatedTo, mountain), (mountain, RelatedTo, large)", (4) "(magma, RelatedTo, volcano), (volcano, RelatedTo, mountain)".

5. Conclusion and future works

In this paper, we explore solving commonsense question answering with explainable reasoning paths. To do this, we propose a PathReasoner that extracts, selects, and learns explainable reasoning paths before answering the questions. The PathReasoner consists of a path finder that performing bi-directional breadth-first searching in a large-scale knowledge graph to retrieve reasoning paths and a hierarchical path learner that softly selecting the paths with intra-path and inter-path features. The reasoning paths facilitate the commonsense question answering, according to the experimental results. Besides, they explain the questions, as shown in the case study. In future work, we will attempt to improve PathReasoner as a more flexible system, where the path finder and the path learner better adapt to each other.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 61976233 and U19A2073), Guangdong Provincial Natural Science Foundation of China (Grant No. 2019B1515120039).

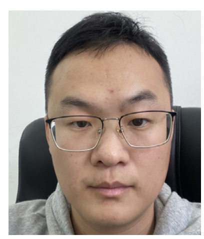
References

[1] J. Berant, A. Chou, R. Frostig, P. Liang, Semantic parsing on freebase from question-answer pairs, in: EMNLP, 2013, pp. 1533–1544.
[2] Q. Cai, A. Yates, Semantic parsing freebase: Towards open-domain semantic parsing, in: SEM, 2013, pp. 328–338.
[3] J. Weston, A. Bordes, S. Chopra, A.M. Rush, B. van Merriënboer, A. Joulin, T. Mikolov, Towards AI-complete question answering: A set of prerequisite toy tasks, in: ICLR, 2016.
[4] M. Richardson, C.J. Burges, E. Renshaw, Mctest: A challenge dataset for the open-domain machine comprehension of text, in: EMNLP, 2013, pp. 193–203.
[5] K.M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, in: NIPS, 2015, pp. 1693–1701.
[6] A. Trischler, T. Wang, X. Yuan, J. Harris, A. Sordoni, P. Bachman, K. Suleman, NewsQA: A machine comprehension dataset, in: ACL, 2017, pp. 191–200.
[7] Z. Yang, P. Qi, S. Zhang, Y. Bengio, W. Cohen, R. Salakhutdinov, C.D. Manning, HotpotQA: A dataset for diverse, explainable multi-hop question answering, in: NIPS, 2018, pp. 2369–2380.
[8] J. Welbl, P. Stenetorp, S. Riedel, Constructing datasets for multi-hop reading comprehension across documents, Trans. Assoc. Comput. Linguist. 6 (2018) 287–302.
[9] A. Talmor, J. Herzig, N. Lourie, J. Berant, CommonsenseQA: A question answering challenge targeting commonsense knowledge, in: NAACL, 2019, pp. 4149–4158.
[10] N. Tandon, B. Dalvi, K. Sakaguchi, P. Clark, A. Bosselut, WIQA: A dataset for "what if…" reasoning over procedural text, in: EMNLP, 2019.
[11] M. Boratko, X. Li, T. O'Gorman, R. Das, D. Le, A. McCallum, ProtoQA: A question answering dataset for prototypical common-sense reasoning, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, Online, 2020, pp. 1122–1136, http://dx.doi.org/10.18653/v1/2020.emnlp-main.85, URL: https://aclanthology.org/2020.emnlp-main.85.
[12] L. Huang, R.L. Bras, C. Bhagavatula, Y. Choi, Cosmos qa: Machine reading comprehension with contextual commonsense reasoning, 2019, arXiv preprint arXiv:1909.00277.
[13] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, in: AAAI, 2017, pp. 4444–4451.
[14] B.Y. Lin, X. Chen, J. Chen, X. Ren, Kagnet: Knowledge-aware graph networks for commonsense reasoning, in: EMNLP-IJCNLP, 2019, pp. 2829–2839.
[15] N.F. Rajani, B. McCann, C. Xiong, R. Socher, Explain yourself! leveraging language models for commonsense reasoning, in: ACL, 2019, pp. 4932–4942.
[16] S. Lv, D. Guo, J. Xu, D. Tang, N. Duan, M. Gong, L. Shou, D. Jiang, G. Cao, S. Hu, Graph-based reasoning over heterogeneous external knowledge for commonsense question answering, in: AAAI, 2020.
[17] A. Asai, K. Hashimoto, H. Hajishirzi, R. Socher, C. Xiong, Learning to retrieve reasoning paths over wikipedia graph for question answering, 2019, arXiv preprint arXiv:1911.10470.
[18] F. Moghimifar, L. Qu, Y. Zhuo, M. Baktashmotlagh, G. Haffari, Cosmo: Conditional Seq2Seq-based mixture model for zero-shot commonsense question answering, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 5347–5359, http://dx.doi.org/10.18653/v1/2020.coling-main.467, URL: https://aclanthology.org/2020.coling-main.467.
[19] Z. Zhang, X. Han, Z. Liu, X. Jiang, M. Sun, Q. Liu, ERNIE: Enhanced language representation with informative entities, in: ACL, 2019, pp. 1441–1451.
[20] W. Zhong, D. Tang, N. Duan, M. Zhou, J. Wang, J. Yin, Improving question answering by commonsense-based pre-training, in: NLPCC, Springer, 2019, pp. 16–28.
[21] S. Sukhbaatar, A. Szlam, J. Weston, R. Fergus, End-to-end memory networks, in: NIPS, 2015, pp. 2440–2448.
[22] Y. Huang, M. Fang, Y. Cao, L. Wang, X. Liang, DAGN: Discourse-aware graph network for logical reasoning, 2021,
[23] R. Li, Z. Jiang, L. Wang, X. Lu, M. Zhao, D. Chen, Enhancing Transformer-based language models with commonsense representations for knowledge-driven machine comprehension, Knowl.-Based Syst. 220 (2021) 106936.
[24] S. Wang, W. Zhong, D. Tang, Z. Wei, Z. Fan, D. Jiang, M. Zhou, N. Duan, Logic-driven context extension and data augmentation for logical reasoning of text, 2021, CoRR abs/2105.03659, URL: https://arxiv.org/abs/2105.03659.

[25] H.J. Levesque, E. Davis, L. Morgenstern, The Winograd schema challenge, in: KR, 2012, pp. 552–561.
[26] A. Bordes, N. Usunier, S. Chopra, J. Weston, Large-scale simple question answering with memory networks, 2015, arXiv, arXiv:1506.02075.
[27] J. Weston, S. Chopra, A. Bordes, Memory networks, 2015, arXiv:1410.3916.
[28] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, R. Socher, Ask me anything: Dynamic memory networks for natural language processing, in: ICML, 2016, pp. 1378–1387.
[29] A. Miller, A. Fisch, J. Dodge, A.-H. Karimi, A. Bordes, J. Weston, Key-value memory networks for directly reading documents, in: EMNLP, 2016, pp. 1400–1409.
[30] Y. Huang, M. Fang, X. Zhan, Q. Cao, X. Liang, L. Lin, REM-net: Recursive erasure memory network for commonsense evidence refinement, 2020, arXiv preprint arXiv:2012.13185.
[31] Q. Chen, F. Ji, H. Chen, Y. Zhang, Improving commonsense question answering by graph-based iterative retrieval over multiple knowledge sources, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 2583–2594, http://dx.doi.org/10.18653/v1/2020.coling-main.232, URL: https://aclanthology.org/2020.coling-main.232.
[32] Y. Cao, M. Fang, D. Tao, BAG: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering, 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), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 357–362, http://dx.doi.org/10.18653/v1/N19-1032, URL: https://aclanthology.org/N19-1032.
[33] M. Tu, G. Wang, J. Huang, Y. Tang, X. He, B. Zhou, Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs, ACL (2019).
[34] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[35] J. Lim, D. Oh, Y. Jang, K. Yang, H. Lim, I know what you asked: Graph path learning using AMR for commonsense reasoning, in: Proceedings of the 28th International Conference on Computational Linguistics, International Committee on Computational Linguistics, Barcelona, Spain (Online), 2020, pp. 2459–2471, http://dx.doi.org/10.18653/v1/2020.coling-main.222, URL: https://aclanthology.org/2020.coling-main.222.
[36] J. Yu, Z. Zhu, Y. Wang, W. Zhang, Y. Hu, J. Tan, Cross-modal knowledge reasoning for knowledge-based visual question answering, Pattern Recognit. 108 (2020) 107563.
[37] G.A. Miller, WordNet: a lexical database for English, Commun. ACM 38 (11) (1995) 39–41.
[38] M. Sap, R.L. Bras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N.A. Smith, Y. Choi, ATOMIC: an atlas of machine commonsense for if-then reasoning, in: AAAI, AAAI Press, 2019, pp. 3027–3035.
[39] H. Zhang, X. Liu, H. Pan, Y. Song, C.W.-K. Leung, ASER: A large-scale eventuality knowledge graph, in: WWW, 2020, pp. 201–211.
[40] T. Zhang, V. Kishore, F. Wu, K.Q. Weinberger, Y. Artzi, BERTscore: Evaluating text generation with BERT, in: ICLR, 2020.
[41] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: NAACL, 2019, pp. 4171–4186.
[42] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, RoBERTa: A robustly optimized BERT pretraining approach, 2019, arXiv preprint arXiv:1907.11692.
[43] Y. Freund, R.E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in: ECCLT, Springer, 1995, pp. 23–37.
[44] M. Assante, L. Candela, D. Castelli, R. Cirillo, G. Coro, L. Frosini, L. Lelii, F. Mangiacrapa, P. Pagano, G. Panichi, F. Sinibaldi, Enacting open science by D4Science, Future Gener. Comput. Syst. 101 (2019) 555–563, http://dx.doi.org/10.1016/j.future.2019.05.063, URL: https://www.sciencedirect.com/science/article/pii/S0167739X1831464X.
[45] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: ICLR, 2015.
[46] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014, pp. 1532–1543.
[47] A.P. Parikh, O. Tackstrom, D. Das, J. Uszkoreit, A decomposable attention model for natural language inference, in: EMNLP, 2016, pp. 2249–2255.
[48] C. Chelba, T. Mikolov, M. Schuster, Q. Ge, T. Brants, P. Koehn, T. Robinson, One billion word benchmark for measuring progress in statistical language modeling, in: INTERSPEECH, 2014, pp. 2635–2639.
[49] J. Thomason, A. Padmakumar, J. Sinapov, N. Walker, Y. Jiang, H. Yedidsion, J. Hart, P. Stone, R.J. Mooney, Improving grounded natural language understanding through human-robot dialog, in: ICRA, 2019, pp. 6934–6941.

Xunlin Zhan is currently a graduate student of the School of Intelligent Systems Engineering at Sun Yat-Sen University. He received the B.Sc. degree from Sun Yat-Sen University in 2019 in Software Engineering. His research interests include natural language processing and commonsense reasoning.

Yinya Huang is currently a Ph.D. student in Computer Science from the School of Intelligent Systems Engineering at Sun Yat-Sen University, advised by Prof. Xiaodan Liang. Her research interests mainly include reasoning-driven natural language processing and graph reasoning. More information can be found on her personal website https://eleanor-h.github.io/.



Xiao Dong is currently a research assistant at Sun Yan-sen University. He received the Master Degree from Shandong Normal University in 2018 supervised by Prof. Huaxiang Zhang. He will start his Ph.D. Candidate at the Sun Yan-sen University in 2021. His research interests include computer vision, machine learning, and data mining. He is a student member of the IEEE.

Qingxing Cao is currently a postdoctoral researcher in the School of Intelligent Systems Engineering at Sun Yat-sen University, working with Prof. Xiaodan Liang. He received his Ph.D. degree from Sun Yat-Sen University in 2019, advised by Prof. Liang Lin. His current research interests include computer vision and visual question answering.

Xiaodan Liang is currently an Associate Professor at Sun Yat-sen University. She was a postdoc researcher in the machine learning department at Carnegie Mellon University, working with Prof. Eric Xing, from 2016 to 2018. She received her Ph.D. degree from Sun Yat-sen University in 2016, advised by Liang Lin. She has published several cutting-edge projects on human-related analysis, including human parsing, pedestrian detection, and instance segmentation, 2D/3D human pose estimation, and activity recognition.