Exploiting Reasoning Chains for Multi-hop Science Question Answering[*]

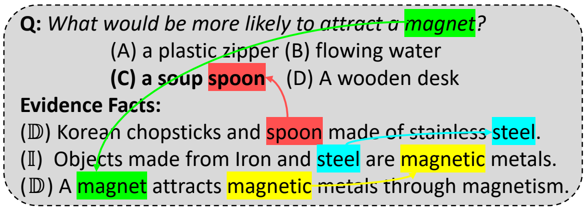
Weiwen Xu, Yang Deng, Huihui Zhang, Deng Cai and Wai Lam
The Chinese University of Hong Kong
{wwxu,ydeng,hhzhang,wlam}@se.cuhk.edu.hk
thisisjcykcd@gmail.com

Abstract

We propose a novel Chain Guided Retriever-reader (CGR) framework to model the reasoning chain for multi-hop Science Question Answering. Our framework is capable of performing explainable reasoning without the need of any corpus-specific annotations, such as the ground-truth reasoning chain, or human-annotated entity mentions. Specifically, we first generate reasoning chains from a semantic graph constructed by Abstract Meaning Representation of retrieved evidence facts. A *Chain-aware loss*, concerning both local and global chain information, is also designed to enable the generated chains to serve as distant supervision signals for training the retriever, where reinforcement learning is also adopted to maximize the utility of the reasoning chains. Our framework allows the retriever to capture step-by-step clues of the entire reasoning process, which is not only shown to be effective on two challenging multi-hop Science QA tasks, namely OpenBookQA and ARC-Challenge, but also favors explainability.

1 Introduction

Question Answering (QA) with external knowledge has gained increasing attention in recent years as it mimics human behavior to first filter out relevant knowledge from massive information. Prior works usually employ a retriever-reader architecture (Chen et al., 2017), where the retriever retrieves top-ranked evidence facts from a large corpus and the reader conducts reasoning with these facts. This architecture works well in single-hop QA, where the answer can be easily inferred with only one evidence fact. However, it is hard to retrieve all necessary evidence facts to confidently answer a complex question requiring multi-hop reasoning (Shao et al., 2021). As shown in Figure 1, multi-hop QA usually involves a sequential nature of evidence facts to form a reasoning chain, including (1) direct facts sharing a semantic relationship with the question or the answer; (2) indirect facts sharing little lexical or semantic overlap but serving an irreplaceable role to infer the answer.



Q: *What would be more likely to attract a magnet?*
(A) a plastic zipper (B) flowing water
(C) a soup spoon (D) A wooden desk
Evidence Facts:
(𝔻) Korean chopsticks and spoon made of stainless steel.
(𝕀) Objects made from Iron and steel are magnetic metals.
(𝔻) A magnet attracts magnetic metals through magnetism.

Figure 1: An example of multi-hop QA with direct (𝔻) and indirect (𝕀) facts to form a reasoning chain.

A common practice for forming such reasoning chains for multi-hop QA is to expand the chain with iterative retrieval (Xiong et al., 2021) or sample from an existing or pre-constructed Knowledge Graph (KG) (Asai et al., 2020; Yasunaga et al., 2021). On one hand, iterative retrieval allows the retriever to capture the evidence-evidence interactions by reformulating the query with newly retrieved evidence fact. However, the retriever would inevitably retrieve partially related facts. Such noise is continuously amplified during the iterative retrieval process, raising concerns about the quality of the reasoning chain. Prior works address this issue by training the retriever against the ground-truth reasoning chain (Yang et al., 2018; Geva et al., 2021). However, such method is less effective when the ground-truth reasoning chain is partially provided (Mihaylov et al., 2018; Ferguson et al., 2020) or not applicable when the ground-truth chain is unavailable (Clark et al., 2018). On the other hand, KG maintains a good growing direction for the reasoning chain. But building a KG usually involves corpus-specific annotations, such as document-level hyperlinks or annotated entity mentions. These limitations make it less applicable

[*] The work described in this paper is substantially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200719)

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 1143–1156
November 7–11, 2021. ©2021 Association for Computational Linguistics

in new domains, where hyperlinks and entities are not prevalent (Xiong et al., 2021).

To address the aforementioned concerns for multi-hop QA, we propose a novel framework, Chain Guided Retriever-reader (CGR), to model the reasoning chains, which is compatible without the ground-truth reasoning chain and applicable to broader textual data. In specific, the proposed framework consists of three components: a retriever, a reasoning chain generator, and a reader. The retriever first iteratively retrieves all possible evidence facts. Then the reasoning chain generator adopts Abstract Meaning Representation (AMR) (Banarescu et al., 2013) to automatically construct a semantic graph that represents the relationship among the retrieved facts. We use AMR because it is not specifically designed for a particular domain and can be adapted to a wide range of sentences (e.g. Science with little named entity). The final generated reasoning chains are supposed to connect the question nodes and the answer nodes on the semantic graph, such that: (i) the generated reasoning chains serving as a byproduct of our framework can support explainability. (ii) the evidence facts on these chains provide a more appropriate context for the reader because they together fill the knowledge gap between the question and the answer. (iii) the reasoning chains can be used as distant supervision signals to train the retriever in case the ground-truth reasoning chain is not available. To achieve these merits, a novel chain-aware loss is proposed to adaptively model the reasoning chains in both supervised and distantly supervised manners. The chain-aware loss not only adopts Reinforcement learning (RL) (Williams, 1992) to maximize the utility of the local information from some certain chains based on the reward from the reader, but also enables the retriever to retrieve indirect evidence facts by considering the global information from all the generated chains. The contributions are summarized as follows:

- Our CGR framework provides a novel formulation to model the reasoning chains, allowing explainable reasoning without the need of any corpus-specific annotations.

- A novel chain-aware loss exploiting both local and global information of reasoning chains is developed to train the retriever, such that the retriever can adapt its retrieval policy to allow high-quality reasoning chains to be generated.

- Experimental results show that CGR can generate reasoning chains to support explainable reasoning and achieve a remarkable improvement on OpenBookQA and ARC-Challenge.

2 Framework

Problem Definition In this work, we tackle the multi-hop Science QA in the form of multi-choices, where each question q_i is associated with J answer choices $a_{ij}, j \in \{1, 2, ..., J\}$. To answer the question, one can refer to an external textual corpus $\mathcal{E}$ for relevant evidence facts. However, since the external corpus is not specifically designed for a particular multi-hop QA task, it does not necessarily contain the relevant evidence facts. Finally, based on the information at hand, our goal is to determine the correct answer.

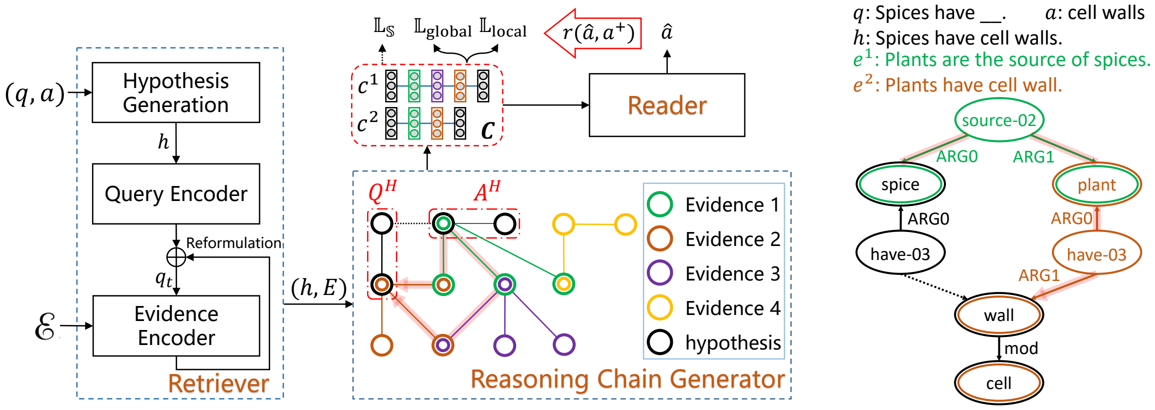
Framework Overview As shown in Figure 2-a, CGR consists of three components: (1) A retriever iteratively retrieves a evidence pool $E = \{e^1, e^2, ...\}$[1] for each question-answer pair (q, a) from an external textual corpus $\mathcal{E}$. (2) A reasoning chain generator first constructs a semantic graph using the fact AMRs in E and then finds all complete reasoning chains $\mathbf{C} = \{\mathbf{c}^1, \mathbf{c}^2, ...\}$ on the semantic graph, where $\mathbf{c}^i$ is a sequence of evidence facts. (3) A reader computes the ranking score of each answer choice, only given the facts $\hat{E} = \{\hat{e}^1, \hat{e}^2, ...\}$ on these reasoning chains as the context. During training time, in addition to the standard reader loss, we propose a novel chain-aware loss that uses $\mathbf{C}$ as distant supervision to train the retriever.

2.1 Retriever

Hypothesis Generation As shown in Figure 2-a, we first generate a hypothesis h for each question-answer pair (q, a) as the initial query for the retriever. Hypothesis generation is to convert a question-answer pair into its declarative form. Such conversion keeps all meaningful contents and maintains a good grammatical structure, which avoids noisy retrieval and allows AMR parser to generate high-quality AMR in Sec. 2.2. We use the rule-based model of Demszky et al. (2018) to generate the hypothesis. For unsolvable cases, we concatenate the question and the answer as the hypothesis.

Iterative Retrieval Taking into account the sequential nature of multi-hop QA, we formulate the retrieval process in an iterative fashion, where the

[1] We omit the subscript ij for simplicity.



(a) Overall Framework

(b) Semantic Graph Construction

Figure 2: Overall architecture of CGR. In the reasoning chain generator, the black dash lines indicate that we discard the edges between question and answer nodes. The pink arrows indicate the generated reasoning chains.

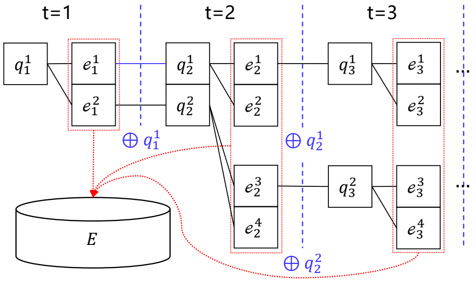


Figure 3: Iterative retrieval process with beam size 2.

query is updated in each iteration, conditioned on the information at hand. We adopt a similar dual-encoder of DPR (Karpukhin et al., 2020). The retrieval process is accomplished by maximum inner-product search (MIPS) between the query representation and the evidence representation:

$$\sigma(q_t, e) = E_q(q_t)^\top E_e(e), e \in \mathcal{E} \qquad (1)$$

where E_q and E_e are BERT$_{\texttt{Base}}$ (Devlin et al., 2019) query encoder and evidence encoder respectively. We take the representation at `[CLS]` as the output text representation. q_t is the reformulated query in each iteration t. We concatenate the query at current iteration with newly retrieved evidence fact to construct the query for the next iteration:

$$q_t = g(q_{t-1}, e_{t-1}) = [q_{t-1}; \texttt{[SEP]}; e_{t-1}] \qquad (2)$$

where $g(\cdot)$ is the reformulation process, $[;]$ is the concatenation. The number of iterations is T. The initial query q_1 is the hypothesis generated above.

As shown in Figure 3, we introduce beam search of size K to retrieve more relevant evidence facts while avoiding the search space from expanding exponentially. After the iterative retrieval, we collect evidence facts retrieved in all iterations to form a large evidence pool E, which is used to build the semantic graph as presented in Sec. 2.2.

2.2 Reasoning Chain Generator

As shown in Figure 2, our reasoning chain generator is a non-parameterized component that generates reasoning chains using the evidence pool. It first relies on AMR to dynamically construct a semantic graph to show the relationship among the facts in the evidence pool, and then generates reasoning chains connecting both question and answer nodes. These reasoning chains serving as a byproduct of our framework can support explainability.

Semantic Graph Construction As depicted in Figure 2-b, AMR nodes are high-level abstractions of concepts conveyed in the corresponding sentence, which capture more semantic information than named entities and can be applied to different domains of sentences. We leverage on the state-of-the-art AMR parser (Cai and Lam, 2020) to generate AMRs $G = \{G^H, G^1, G^2, ...\}$ for the hypothesis and its corresponding evidence pool, where G^H, G^i are the AMR of the hypothesis and the i^{th} fact in the evidence pool. The construction procedures are given as follows:

Nodes: We reuse most of the concepts found in G as the nodes for our semantic graph except for some over-general concepts (e.g. `(p/planet:name(n/name:op1"Earth"))`, node `n/name` is an over-general concept). Fortunately, such nodes always have non-node attribute (e.g. `Earth` of `n/name`) that shows the specific referent. Therefore, we replace concepts with their attributes as the nodes in the semantic graph if applicable.

Inner-AMR Edges: We split the nodes of G^H into question nodes Q^H and answer nodes A^H to represent the concepts conveyed in the question and the answer respectively. As one question is provided with J answer choices, where we can generate J hypothesis AMRs accordingly for each question. We take the shared nodes of J hypothesis AMRs as Q^H, while the remaining as A^H:

$$Q_{ij}^H = \cap_{j=1}^{J} \{v | v \in G_{ij}^H\}, A_{ij}^H = \{v | v \in G_{ij}^H, v \notin Q_{ij}^H\} \tag{3}$$

We preserve the edges in G^H as edges in our semantic graph except for the edges between Q^H and A^H to guarantee the reasoning chains are spotted outside G^H. All edges in G^i are preserved.

Inter-AMR Edges: We add an edge between nodes of the same concept, except the root of any AMR, because these roots are mostly verbs, which may create unwanted shortcuts (e.g. `have-03` in Figure 2-b). Duplicate nodes will be unified into one node, and all edges connecting those duplicate nodes are redirected to the unified node.

Reasoning Chain Generation As shown in Figure 2-a, a reasoning chain is a sequence of evidence facts that logically connect the question with the answer, while our constructed semantic graph consists of nodes in the form of AMR concepts. To tackle the mismatch, we preserve the source of each concept to show which evidence fact it comes from. During generation, we first apply depth first search on our constructed semantic graph to find all completed paths that start from one of the question nodes, pass multiple nodes outside G^H, and end at one of the answer nodes. We then map the nodes to their source evidence facts and thus form evidence-level reasoning chains $\mathbf{C}$. We find the evidence-level reasoning chain is effective enough to capture the relationships among evidence facts and leave a more fine-grained concept-level graph in future work.

2.3 Reader

Existing works (Chen et al., 2017) typically package all the top-retrieved facts as the context for each question-answer pair, while we use the evidence facts $\hat{E}$ only on the generated reasoning chains.

$$\hat{E} = \cup_{i=1}^{l} \{e | e \in \mathbf{c}^i\} \tag{4}$$

We concatenate the question, the answer and all the evidence facts in $\hat{E}$ as the reader input, where `[CLS]` is inserted at the beginning of the sequence, and `[SEP]` are separators among the three texts. The output `[CLS]` representation is fed into a classification layer, which is a linear transformation for computing the score for each question-answer pair.

2.4 Chain-Aware Loss

2.4.1 Reasoning Chain Modeling

Similar to our iterative retrieval, we model the reasoning chain in an iterative fashion. The probability of generating one reasoning chain $\mathbf{c} = \{e_t\}_{t=1}^{l}$ is:

$$p(\mathbf{c}|h) = \prod_{t=1}^{l} \frac{\exp(\sigma(q_t, e_t))}{\exp(\sigma(q_t, e_t)) + \sum\limits_{e \in \mathcal{B}_t^-} \exp(\sigma(q_t, e))} \tag{5}$$

where $\mathcal{B}_t$ denotes all evidence facts at the t-th iteration and $\mathcal{B}_t^- = \mathcal{B}_t \backslash e_t$ denotes the in-batch negative examples for e_t, q_t is the reformulated query at the t-th iteration from Eq. (2) with $q_1 = h$.

Based on this modeling, we propose a novel chain-aware loss to train our retriever, which enables both **supervised training** with a ground-truth reasoning chain as well as **distantly supervised training** with generated reasoning chains.

2.4.2 Supervised Training

Some datasets (Mihaylov et al., 2018; Khot et al., 2020) annotate or partially annotate the ground-truth reasoning chain, which is a good supervision to train the retriever. We take the ground-truth reasoning chain as one part of our chain-aware loss if the dataset provided. Specifically, let $\mathbf{c}^+$ be an ordered ground-truth reasoning chain and h^+ be the generated hypothesis corresponding to the correct answer a^+. The supervised chain loss is defined as:

$$\mathbb{L}_{\mathbb{S}} = -\log p(\mathbf{c}^+ | h^+) \tag{6}$$

2.4.3 Distantly Supervised Training

The ground-truth reasoning chain is quite corpus-dependent and expensive to obtain, which is not applicable in many multi-hop QA tasks (Clark et al., 2018). To handle this, we leverage the generated reasoning chains from Sec. 2.2 as weak supervision signals to facilitate the distantly supervised training of the retriever. Since there are likely multiple generated reasoning chains for the same hypothesis, we elaborate two losses, namely **Local Chain Loss** and **Global Chain Loss**, to model the local chain information (in a certain reasoning chain) and the global chain information (across all the reasoning chains), respectively.

Local Chain Loss (MLE) The local chain information is modeled in a *bidirectional sequence* manner that maximizes the likelihood (MLE) between the current evidence fact with all previous evidence facts in the reasoning chain. Specifically, we sample up to N chains $\{\hat{\mathbf{c}}^i\}^N$ from the generated reasoning chains **C** corresponding to h^+. The forward part of MLE loss is defined by traversing each $\hat{\mathbf{c}}^i$ forwardly:

$$\overrightarrow{\mathbb{L}_{\text{mle}}} = -\frac{1}{N}\sum\nolimits_{i=1}^{N} \log p(\overrightarrow{\hat{\mathbf{c}}^i}|h^+) \tag{7}$$

Sometimes, it may be difficult to retrieve multi-hop evidence facts from a forward direction when the question is rather confusing. Similarly, we define the backward loss $\overleftarrow{\mathbb{L}_{\text{mle}}}$ that traverses each $\hat{\mathbf{c}}^i$ in a reversed order. Then the MLE loss is:

$$\mathbb{L}_{\text{mle}} = \overrightarrow{\mathbb{L}_{\text{mle}}} + \overleftarrow{\mathbb{L}_{\text{mle}}} \tag{8}$$

Local Chain Loss (RL) Despite the independent training of the retriever and the reader, the reader is supposed to benefit from good reasoning chains. To bridge such gap, we pass the information of how the reasoning chains affect the reader performance to the retriever via Reinforcement Learning (RL) (Williams, 1992), such that the retriever can adapt its retrieval policy accordingly and thus allow high-quality reasoning chains to be generated. Specifically, given the question-answer pair as well as the corresponding reasoning chains, the *action* follows the same reasoning chains sampled above. The *policy* defines the probability of bidirectionally generating those reasoning chains using Eqn. 5 and the *reward* defines the correctness of the reader prediction. We use the following RL loss to update the parameters of the retriever via policy gradient:

$$\begin{aligned}\mathbb{L}_{\text{rl}} = -\frac{1}{N}&\sum\nolimits_{i=1}^{N}[r(\hat{a}, a^+) - \bar{r}]\cdot \\ &[\log p(\overrightarrow{\hat{\mathbf{c}}^i}|h^+) + \log p(\overleftarrow{\hat{\mathbf{c}}^i}|h^+)]\end{aligned} \tag{9}$$

where $\hat{a}$ is the predicted answer, $r(\cdot,\cdot)$ is the reward implemented as the $0/1$ indicator function. $\bar{r}$ is the average reward in a mini-batch as the bias term.

Finally, the local chain loss is the combination of MLE and RL loss:

$$\mathbb{L}_{\text{local}} = \mathbb{L}_{\text{mle}} + \mathbb{L}_{\text{rl}} \tag{10}$$

Global Chain Loss A reasoning chain usually involves indirect facts that are only related to the direct facts while share little semantic overlap with the hypothesis. Such indirect facts are unlikely to be retrieved if we fail to retrieve their corresponding direct facts. To handle this, we compute a global representation of reasoning chains by averaging the representations of all evidence facts in these chains and propose a *global chain loss* to maximize the likelihood between the hypothesis and the global chain information:

$$\begin{aligned}\mathbb{L}_{\text{global}} &= -\log \frac{\psi(\hat{E},h^+)}{\psi(\hat{E},h^+)+\sum\limits_{E\in\mathcal{B}^-}\psi(E,h^+)} \\ \psi(\hat{E},h^+) &= \exp(\tfrac{1}{|\hat{E}|}\sum\nolimits_{e\in\hat{E}}\sigma(h^+,e))\end{aligned} \tag{11}$$

where we use the similar in-batch negative examples $\mathcal{B}^- = \mathcal{B} \setminus \hat{E}$ to train the global chain loss as well. $\mathcal{B}$ denotes the collection of all evidence facts in current mini-batch and $\hat{E}$ is the set of evidence facts selected by Eqn. 4 for the current hypothesis h^+.

2.5 Training & Inference

We use the supervised chain-aware loss $\mathbb{L}_{\mathbb{S}}$ and the distantly supervised chain-aware loss $\mathbb{L}_{\mathbb{D}} = \mathbb{L}_{\text{local}} + \mathbb{L}_{\text{global}}$ to train the retriever and the standard Cross-Entropy loss $\mathbb{L}_{\text{reader}}$ between the reader prediction and the correct answer choice to train the reader. The final training objective is:

$$\mathbb{L} = \mathbb{L}_{\text{reader}} + \mathbb{L}_{\mathbb{S}} + \mathbb{L}_{\mathbb{D}} \tag{12}$$

During inference, each question-answer pair follows the same pipeline of our retriever, reasoning chain generator, and reader to get its ranking score. The top-ranked choice is chosen as the output.

3 Experimental Setup

Datasets: We evaluate the effectiveness of our CGR framework on two multi-hop science QA datasets: OpenBookQA (Mihaylov et al., 2018) and ARC-Challenge (Clark et al., 2018), where the ground-truth reasoning chain is either partially provided or not provided. OpenBookQA and ARC-Challenge provide their corresponding leaderboards with train, develop and test sets publicly available. Following AllenAI (2019), we combine the training set of OpenBookQA (4957), ARC-Easy (2251), ARC-Challenge (1119) and *RegLivEnv* (665) as the final training set of ARC-Challenge task. The data splits is shown in Table 1. OpenBookQA annotates one evidence fact on the ground-truth reasoning chain for each question-answer pair, which is used in $\mathbb{L}_{\mathbb{S}}$. ARC-Challenge does not provide the ground-truth reasoning chain, where $\mathbb{L}_{\mathbb{S}}$ is eliminated from the final training objective. The textual corpus is ARC Corpus (Clark

	Train	Dev	Test
OpenBookQA	4957	500	500
ARC-Challenge	8992	299	1172

Table 1: Number of instances in each dataset.

Methods	Additional KG	Output Chains	Test Acc.
PG	✓	✓	81.8
AMR-SG	×	×	81.6
DPR	×	×	80.8
MHGRN	✓	✓	80.6
KF-SIR	×	×	80.0
AristoRoBERTa	×	×	77.8
+ CGR	×	✓	**82.4**

Table 2: Test accuracy on OpenBookQA. Methods that use additional KG or can output reasoning chains are ticked respectively.

Methods	Additional KG	Output Chains	Test Acc.
AMR-SG	×	×	68.94
FreeLB	×	×	67.75
arcRoberta ♠	×	×	67.15
xlnet+Roberta ♠	×	×	67.06
AristoRoBERTa	×	×	66.47
+ CGR	×	✓	**69.20**

Table 3: Test accuracy on ARC-Challenge. ♠ are unpublished methods.

et al., 2018) for both two tasks, consisting of 14M science facts. Moreover, OpenBookQA provides an accompanying open-book with 1326 science facts, which are highly related to the questions in this dataset. Therefore, we performed an extra retrieval in the open-book in the first iteration.

Implementation: We use AristoRoBERTa (AllenAI, 2019) as our reader and Sentence-BERT (Reimers and Gurevych, 2019) as our retriever. As indicated by Lewis et al. (2020), training the evidence encoder E_e that requires periodic updates of the evidence index is costly and does little improvement to the performance. Therefore, we fix the evidence encoder and cache all evidence representations offline for efficiency purposes. We set the beam size K to 10 and the total iteration step T to 2, resulting in an average size of evidence pool to be 78 and 53 for OpenBookQA and ARC-Challenge respectively. We then select facts for the reader with maximum evidence size of 15 and 20 respectively. The number of sampled chains N is set to 1. [2] More details and analysis can be found in Appendix A.1 and A.2.

Comparison Methods: For fair comparison, we compare CGR with recently published methods that use similar power of pretrained models, including five textual knowledge based methods: AristoRoBERTa (AllenAI, 2019), KF-SIR (Banerjee and Baral, 2020), FreeLB (Zhu et al., 2020), DPR (Karpukhin et al., 2020), AMR-SG (Xu et al., 2021) and another two methods leveraging on an additional knowledge graph (Speer et al., 2017): PG (Wang et al., 2020), and MHGRN (Feng et al., 2020).

4 Results

4.1 QA Performance

OpenBookQA: Table 2 shows the comparison results of OpenBookQA. Our CGR significantly improves over the baseline AristoRoBERTa with 4.6 accuracy score. Meanwhile, CGR can also provide the reasoning chains as an additional output that reflect the step-by-step reasoning process to infer the answer, making the QA process explainable. When compared to recently published methods, we find that CGR can also surpass methods leveraging on additional KG. It suggests that textual knowledge resource is still under-investigated, where the gap between the query and indirect fact is one of the issues that restricts the retriever performance for multi-hop QA. UnifiedQA (Khashabi et al., 2020) and T5 3B (Raffel et al., 2020) are two extremely large models (with 30x more parameters than other models), which are not fair for comparison.

ARC-Challenge: We implement CGR on another task: ARC-Challenge, where the ground-truth reasoning chain is not available. As shown in Table 3, our CGR significantly improves the baseline AristoRoBERTa with 2.73 accuracy score, which demonstrates the effectiveness of CGR in generating and modeling the reasoning chain in a more general manner. Notably, CGR achieves a new state-of-the-art performance in this challenging task in a computationally practicable setting.

Ablation Study: Table 4 shows our ablation study on the composition of the training objectives both in the presence or absence of the ground-truth reasoning chain on OpenBookQA. We can observe

[2] Our code is available at: https://github.com/wwxu21/CGR

Methods	w/o $\mathbb{L}_{\mathbb{S}}$	w/ $\mathbb{L}_{\mathbb{S}}$
CGR	79.70±0.33	**81.70±0.49**
- $\mathbb{L}_{\text{global}}$	78.60±0.28	80.95±0.30
- $\mathbb{L}_{\text{local}}$ (MLE)	79.10±0.30	80.55±0.52
- $\mathbb{L}_{\text{local}}$ (RL)	78.70±0.36	80.80±0.22
- $\mathbb{L}_{\text{local}}$ (Both)	78.40±0.37	81.05±0.30
- $\mathbb{L}_{\mathbb{D}}$	78.20±0.37	80.55±0.38

Table 4: Ablation study on the composition of the training objectives on OpenBookQA. Accuracy (mean± standard deviation) are computed over 4 runs.

IR Methods	Acc.	Dir.	Ind.	Com.
TF-IDF	52.4	3.72	0.38	0.46
Dense Vector	54.6	4.68	0.40	0.68
Iterative Retrieval	**63.6**	5.00	0.48	0.70
Chain Generator	60.2	**5.24**	**0.96**	**0.74**

Table 5: Automatic and Human Evaluations of the IR performance on OpenBookQA.

the same performance trend under two scenarios. First, we observe a degradation of QA performance when removing $\mathbb{L}_{\text{global}}$. As $\mathbb{L}_{\text{global}}$ provides a rough idea of the reasoning chain, it reduces the difficulty to retrieve indirect facts. Moreover, $\mathbb{L}_{\text{global}}$ is still important even the ground-truth evidence chain is present because it improves the generalization of our framework to retrieve other reasoning chains that can answer the question rather than overfitting to the ground-truth reasoning chain. Second, discarding $\mathbb{L}_{\text{local}}$ also casts a negative impact on the QA performance. $\mathbb{L}_{\text{local}}$, on the other hand, is a fine-grained modeling of evidence-evidence interactions. It is effective to distinguish the correct answer because the incorrect ones would get a relatively low probability to form the reasoning chain. Third, discarding both global and local chain losses results in more severe performance degradation, which demonstrates the necessity of our modeling for the reasoning chains both globally and locally.

4.2 Iterative Retriever Performance

As mentioned, one of our major contributions is to form reasoning chains that capture both direct and indirect evidence facts. To evaluate the quality of the retrieved facts, we conduct both automatic and human evaluations. As OpenBookQA provides one ground-truth evidence fact, we use the retrieval accuracy (Acc.) as our automatic evaluation. For human evaluation, we evaluate the quality from three aspects: (1) *Directly-Related (Dir.):* The evidence is a direct fact and is useful to answer the question. (2) *Indirectly-Related (Ind.):* The evidence is an indirect fact and is useful to answer the question. (3) *Completeness (Com.):* All evidence facts in $\hat{E}$ can together form a real reasoning chain that completely fills the knowledge gap between the question and the answer. We randomly sample 50 questions and evaluate the quality of evidence facts corresponding to the correct answer choice, where each fact contributes 1 score if it meets the requirement of *Dir.* and *Ind.* (ranging from 0 to 15), and all evidence facts in $\hat{E}$ contribute 1 score if they together meet *Com.* (ranging from 0 to 1).

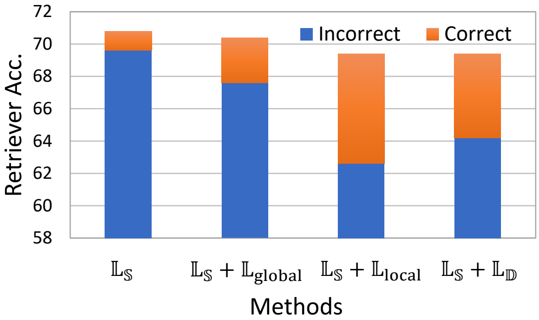


Figure 4: Retriever accuracy on OpenBookQA. The ground-truth evidence in the context of (1) the correct answer choice; (2) any three incorrect answer choices.

As shown in Table 5, we conduct evaluations on four Information Retrieval (IR) methods. Among those IR methods, Dense Vector retriever is effective in finding direct facts than word-match based retriever (TF-IDF) but faces the same limitation in finding indirect facts. Iterative Retrieval can remedy this to some extent, but it is a relatively loose restriction, where the retrieved facts can be biased to some particular facts. Surprisingly, our Reasoning Chain Generator significantly improves the recall of retrieving indirect facts with *Ind.* almost doubled. Though Reasoning Chain Generator may hurt *Acc.*, the improvements on both *Dir.* and *Ind.* show that it can still find alternative facts from the textual corpus to form the reasoning chain.

5 Discussions on Explainability

5.1 Effect of Chain Modeling

We plot the accuracy of the ground-truth evidence fact retrieved either by the hypothesis corresponding to the correct answer and the hypotheses corresponding to incorrect answers in Figure 4. Firstly, though the main purpose of $\mathbb{L}_{\text{global}}$ is to improve the generalization ability of our framework, it can also slightly reduce the retrieval accuracy for incorrect answers with little hurt to the correct answer.

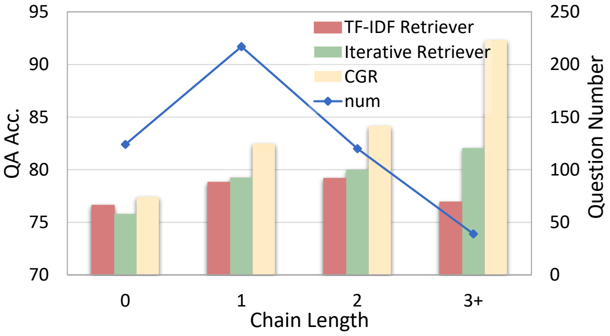


Figure 5: QA performance in terms of reasoning chain length on OpenBookQA. TF-IDF and iterative retriever only use retriever-reader architecture.

Question: *A person wants to be able to have more natural power in their home. They choose to cease using a traditional electric company to source this electricity, and so decide to install* (A) sun grafts (B) sunlight shields **(C) panels collecting sunlight** (D) solar bees
Useful facts retrieved by iterative retrieval: [1] A solar panel converts sunlight into electricity.
Additional facts from reasoning chain generator: [2] Solar energy is a renewable resource. [3] Such renewable resources are called, natural resources.
Reasoning Chain: Question $\xrightarrow{\texttt{natural-03}}$ [3] $\xrightarrow{\texttt{renew-01}}$ [2] $\xrightarrow{\texttt{solar}}$ [1] $\xrightarrow{\texttt{panel}}$ (C)

Table 6: Case study on OpenBookQA.

Secondly, $\mathbb{L}_{\text{local}}$ significantly reduces the retrieval accuracy for incorrect answers. It makes the incorrect answers less deceptive and thus makes it much easier for the reader to distinguish the correct answer, which is concise with the results in Table 4. Thirdly, the combination of the two chain-aware losses may affect the retriever performance marginally, but in terms of overall QA performance, they obviously bring benefits.

5.2 Scalability on Chain length

We plot the QA performance in terms of different chain lengths on OpenBookQA in Figure 5. As we may generate multiple reasoning chains with different lengths for each question-answer pair, we define the length of the reasoning chain for each question as the most frequent length with respect to the correct answer. Length of 0 indicates that we cannot generate any reasoning chain for these questions, because (1) some decisive evidence facts are missing in our textual corpus, and (2) the performance of the AMR parser also limits the upper bound of our reasoning chain generator. Meanwhile, such cases are also difficult for TF-IDF and iterative retriever, which should be highly considered in the future. Apart from this, we can observe a comprehensive improvement of CGR in handling multi-hop QA. Such improvement is much more obvious when the chain length becomes larger, which is commonly hard for reasoning. This demonstrates the effectiveness of our explicit modeling for the reasoning chain, especially when the question is more complex.

5.3 Case Study on Reasoning

We show how our generated reasoning chain can support explainable reasoning in Table 6. Iterative retrieval can retrieve the first evidence fact as it shares critical semantic information `panel` with the choice `C`. However, it fails to retrieve the second and the third evidence fact because (1) the second one is an indirect fact sharing little semantic overlap with either the question or the answer choice, and (2) the third one though serves as a direct fact, it shows a relatively low similarity with the question due to the massive information conveyed in the question. On the other hand, our reasoning chain is able to discover evidence facts that fail to be retrieved by an iterative retriever and form a reasoning chain with AMR concepts as anchors. As the incorrect answers are not likely to form reasoning chains, the evidence facts on the reasoning chains are highly discriminative and can effectively support the reader to select the correct answer. More cases can be found in Appendix A.3.

6 Related Work

Multi-hop QA: Multi-hop QA is a challenging task as it requires gathering multiple evidence facts, especially indirect facts, to form a reasoning chain. Early attempts mostly rely on iterative retrieval. For example,Yadav et al. (2019) extract evidence facts in consideration of their relevance, overlap and coverage. Banerjee et al. (2019); Yadav et al. (2020) reformulate their queries with unused words in the last iteration. However, these methods may retrieve irrelevant facts as the query grow biased to unimportant words. As some recent QA datasets annotate the ground-truth reasoning chain (Yang et al., 2018; Mihaylov et al., 2018), they enable training supervised classifier to identify the correct evidence facts (Nie et al., 2019; Tu et al., 2020). It is a good step to control the quality of reasoning chains, but still remains an issue when the ground-truth reasoning chain is not available (Clark et al., 2018). Other works explore the effectiveness of KG by either automatically constructing the graph using named entity or semantic role labeling (Qiu et al., 2019; Bosselut et al., 2019; Fang et al., 2020; Chen et al., 2019) or resorting to existing KG (Saxena et al., 2020; Zhang et al., 2020; Yasunaga et al., 2021). Despite the high precision of those KGs, they are known to suffer from sparsity in existing KG (Zhao et al., 2020), where complex reasoning chains are unlikely to be covered by the closed-form relations in KG (Lei et al., 2020).

Dense-Vector Retriever: In contrast to term-based retriever implemented with TF-IDF or BM25 (Chen et al., 2017; Wang et al., 2018), dense-vector retriever has received increasing attention as it captures the semantic matching beyond simple word overlap and can be trained along with the reader (Zhu et al., 2021). It has been reported to outperform term-based methods in many open-domain QA tasks (Das et al., 2019; Karpukhin et al., 2020; Min et al., 2021), including those on multi-hop QA (Asai et al., 2020; Xiong et al., 2021).

AMR: AMR has been successfully coupled with many natural language processing tasks in explicit reasoning, such as summarization (Liao et al., 2018), event detection (Li et al., 2015), machine translation (Song et al., 2019), and symbolic QA (Kapanipathi et al., 2020). Comparing to named entity (Shao et al., 2020), we use AMR as our graph annotation because it is not specifically designed for a particular domain and can be adapted to a wide range of textual data.

7 Conclusion

We propose a novel Chain Guided Retriever-reader framework for multi-hop QA. Our modeling for the reasoning chains is effective to find both direct and indirect facts and is less likely to introduce noise. Moreover, our framework is corpus-independent and is capable of handling the setting without any ground-truth annotations. Further analysis and discussions also elucidate some of the inner workings of our framework while maintaining the explainability at the same time.

References

AllenAI. 2019. Aristorobertav7. In *AristoRoBERTAv7*.

Akari Asai, Kazuma Hashimoto, Hannaneh Hajishirzi, Richard Socher, and Caiming Xiong. 2020. Learning to retrieve reasoning paths over wikipedia graph for question answering. In *International Conference on Learning Representations*.

Laura Banarescu, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer, and Nathan Schneider. 2013. Abstract Meaning Representation for sembanking. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 178–186, Sofia, Bulgaria. Association for Computational Linguistics.

Pratyay Banerjee and Chitta Baral. 2020. Knowledge fusion and semantic knowledge ranking for open domain question answering. *arXiv preprint arXiv:2004.03101*.

Pratyay Banerjee, Kuntal Kumar Pal, Arindam Mitra, and Chitta Baral. 2019. Careful selection of knowledge to solve open book question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6120–6129, Florence, Italy. Association for Computational Linguistics.

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

Deng Cai and Wai Lam. 2020. AMR parsing via graph-sequence iterative inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1290–1301, Online. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Jifan Chen, Shih-ting Lin, and Greg Durrett. 2019. Multi-hop question answering via reasoning chains. *arXiv preprint arXiv:1910.02610*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, and Andrew McCallum. 2019. Multi-step retriever-reader interaction for scalable open-domain question answering. In *International Conference on Learning Representations*.

Dorottya Demszky, Kelvin Guu, and Percy Liang. 2018. Transforming question answering datasets into natural language inference datasets. *arXiv preprint arXiv:1809.02922*.

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

Yuwei Fang, Siqi Sun, Zhe Gan, Rohit Pillai, Shuohang Wang, and Jingjing Liu. 2020. Hierarchical graph network for multi-hop question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8823–8838, Online. Association for Computational Linguistics.

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

James Ferguson, Matt Gardner, Hannaneh Hajishirzi, Tushar Khot, and Pradeep Dasigi. 2020. IIRC: A dataset of incomplete information reading comprehension questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1137–1147, Online. Association for Computational Linguistics.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies.

Pavan Kapanipathi, Ibrahim Abdelaziz, Srinivas Ravishankar, Salim Roukos, Alexander Gray, Ramon Astudillo, Maria Chang, Cristina Cornelio, Saswati Dana, Achille Fokoue, et al. 2020. Question answering over knowledge bases by leveraging semantic parsing and neuro-symbolic reasoning. *arXiv preprint arXiv:2012.01707*.

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

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

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

Wenqiang Lei, Gangyi Zhang, Xiangnan He, Yisong Miao, Xiang Wang, Liang Chen, and Tat-Seng Chua. 2020. Interactive path reasoning on graph for conversational recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2073–2083.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Xiang Li, Thien Huu Nguyen, Kai Cao, and Ralph Grishman. 2015. Improving event detection with Abstract Meaning Representation. In *Proceedings of the First Workshop on Computing News Storylines*, pages 11–15, Beijing, China. Association for Computational Linguistics.

Kexin Liao, Logan Lebanoff, and Fei Liu. 2018. Abstract Meaning Representation for multi-document summarization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1178–1190, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Sewon Min, Jordan Boyd-Graber, Chris Alberti, Danqi Chen, Eunsol Choi, Michael Collins, Kelvin Guu, Hannaneh Hajishirzi, Kenton Lee, Jennimaria Palomaki, Colin Raffel, Adam Roberts, Tom Kwiatkowski, Patrick Lewis, Yuxiang Wu, Heinrich Küttler, Linqing Liu, Pasquale Minervini, Pontus Stenetorp, Sebastian Riedel, Sohee Yang, Minjoon Seo, Gautier Izacard, Fabio Petroni, Lucas Hosseini, Nicola De Cao, Edouard Grave, Ikuya Yamada, Sonse Shimaoka, Masatoshi Suzuki, Shumpei Miyawaki, Shun Sato, Ryo Takahashi, Jun Suzuki, Martin Fajcik, Martin Docekal, Karel Ondrej, Pavel Smrz, Hao Cheng, Yelong Shen, Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao, Barlas Oguz, Xilun Chen, Vladimir Karpukhin, Stan Peshterliev, Dmytro Okhonko, Michael Schlichtkrull, Sonal Gupta, Yashar Mehdad, and Wen tau Yih. 2021. Neurips 2020 efficientqa competition: Systems, analyses and lessons learned.

Yixin Nie, Songhe Wang, and Mohit Bansal. 2019. Revealing the importance of semantic retrieval for machine reading at scale. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2553–2566, Hong Kong, China. Association for Computational Linguistics.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Apoorv Saxena, Aditay Tripathi, and Partha Talukdar. 2020. Improving multi-hop question answering over knowledge graphs using knowledge base embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4498–4507, Online. Association for Computational Linguistics.

Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu. 2020. Is Graph Structure Necessary for Multi-hop Question Answering? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7187–7192, Online. Association for Computational Linguistics.

Nan Shao, Yiming Cui, Ting Liu, Shijin Wang, and Guoping Hu. 2021. Memory augmented sequential paragraph retrieval for multi-hop question answering.

Linfeng Song, Daniel Gildea, Yue Zhang, Zhiguo Wang, and Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics*, 7:19–31.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

Ming Tu, Kevin Huang, Guangtao Wang, Jing Huang, Xiaodong He, and Bowen Zhou. 2020. Select, answer and explain: Interpretable multi-hop reading comprehension over multiple documents. In *AAAI*, pages 9073–9080.

Peifeng Wang, Nanyun Peng, Filip Ilievski, Pedro Szekely, and Xiang Ren. 2020. Connecting the dots: A knowledgeable path generator for commonsense question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4129–4140, Online. Association for Computational Linguistics.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018. R 3: Reinforced ranker-reader for open-domain question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256.

Wenhan Xiong, Xiang Li, Srini Iyer, Jingfei Du, Patrick Lewis, William Yang Wang, Yashar Mehdad, Scott Yih, Sebastian Riedel, Douwe Kiela, and Barlas Oguz. 2021. Answering complex open-domain questions with multi-hop dense retrieval. In *International Conference on Learning Representations*.

Weiwen Xu, Huihui Zhang, Deng Cai, and Wai Lam. 2021. Dynamic semantic graph construction and reasoning for explainable multi-hop science question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1044–1056, Online. Association for Computational Linguistics.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2578–2589, Hong Kong, China. Association for Computational Linguistics.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2020. Unsupervised alignment-based iterative evidence retrieval for multi-hop question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4514–4525, Online. Association for Computational Linguistics.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Michihiro Yasunaga, Hongyu Ren, Antoine Bosselut, Percy Liang, and Jure Leskovec. 2021. Qagnn: Reasoning with language models and knowledge graphs for question answering. *arXiv preprint arXiv:2104.06378*.

Yao Zhang, Xu Zhang, Jun Wang, Hongru Liang, Adam Jatowt, Wenqiang Lei, and Zhenglu Yang. 2020. Gmh: A general multi-hop reasoning model for kg completion. *arXiv preprint arXiv:2010.07620*.

Chen Zhao, Chenyan Xiong, Xin Qian, and Jordan Boyd-Graber. 2020. Complex factoid question answering with a free-text knowledge graph. In *Proceedings of The Web Conference 2020*, pages 1205–1216.

Chen Zhu, Yu Cheng, Zhe Gan, Siqi Sun, Tom Goldstein, and Jingjing Liu. 2020. Freelb: Enhanced adversarial training for natural language understanding. In *International Conference on Learning Representations*.

Fengbin Zhu, Wenqiang Lei, Chao Wang, Jianming Zheng, Soujanya Poria, and Tat-Seng Chua. 2021. Retrieving and reading: A comprehensive survey on open-domain question answering. *arXiv preprint arXiv:2101.00774*.

A Appendix

A.1 Implementation

For OpenBookQA, we implement our CGR framework only in the last fine-tuning process as indicated in AllenAI (2019), where only OpenBookQA dataset is used. The initial learning rate is 9e-6, the batch size is 4 and the max sequence length is 256. For ARC-Challenge, we implement our CGR framework on the combination of all above training sets. The initial learning rate, the batch size and the max sequence length are 1e-5, 6, and 384 respectively. We use grid search to find optimal hyper-parameters, where the learning rate is chosen from {5e-6,8e-6,9e-6,1e-5,1.1e-5, 1.2e-5,1.5e-5}, the batch size is chosen from {4,6,8,12,16}, beam size K is chosen from {5,10,15,20} and iteration step T is chosen from {1,2,3}.

We introduce 110M parameters of our retriever in addition to 355M of our reader. We run all experiments on one TITAN RTX card, which takes about 2 hour and 8 hours to complete the training of OpenBookQA and ARC-Challenge respectively.

A.2 Effect of Beam size and Iteration Step

We vary two hyper-parameters K and T to show their effects on OpenBookQA. As depicted in Figure 6, the model with $T = 1$ has a relatively lower performance than the other two models because it suffers from a low recall of relevant evidence facts, which also explains why it benefits more from a larger K. Moreover, model with $T = 2$ performs better than model with $T = 3$. It indicates most of the questions can be solved with a reasoning chain of length 2, which is consistent with the construction of this dataset. In addition, models with $T > 1$ reaches the top at $K = 10$. This might be due to more noisy retrievals in a larger evidence pool.

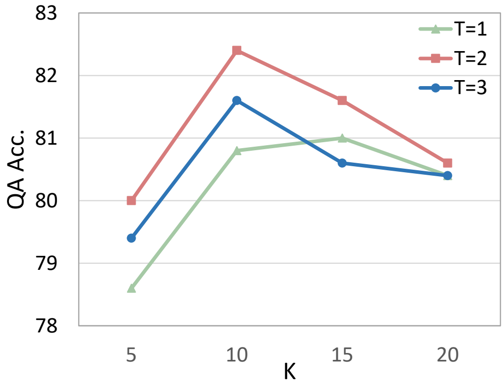


Figure 6: Effect of hyper-parameter Beam size (K) and Iteration Step (T) on OpenBookQA.

A.3 Case Study

More case studies can be found in Table 7.

(a) Case Study 1

Question:	*Which requires energy to move?* **(A) weasel** (B)willow (C) mango (D) poison ivy
Related Evidence Facts:	[1] An animal requires energy to move. [2] Predator is a animal. [3] A weasels food chain is a predator.
Reasoning Chain:	Question $\xrightarrow{\texttt{energy}}$ [1] $\xrightarrow{\texttt{animal}}$ [2] $\xrightarrow{\texttt{predator}}$ [3] $\xrightarrow{\texttt{weasel}}$ (A)

(b) Case Study 2

Question:	*A positive effect of burning biofuel is* (A) shortage of crops for the food supply (B) an increase in air pollution **(C) powering the lights in a home** (D) deforestation in the amazon to make room for crops
Related Evidence Facts:	[1] Biofuel is used to produce electricity by burning. [2] Some light bulbs convert electricity into light and heat energy.
Reasoning Chain:	Question $\xrightarrow{\texttt{biofuel}}$ [1] $\xrightarrow{\texttt{electricity}}$ [2] $\xrightarrow{\texttt{light}}$ (C)

(c) Case Study 3

Question:	*An example of conservation is avoiding the use of* **(A) gasoline** (B) air (C) snow (D) clothes
Related Evidence Facts:	[1] An example of conservation is not using fossil fuel. [2] Gasoline is a fuel mixture.
Reasoning Chain:	Question $\xrightarrow{\texttt{conserve-01}}$ [1] $\xrightarrow{\texttt{fuel}}$ [2] $\xrightarrow{\texttt{gasoline}}$ (A)

(d) Case Study 4

Question:	*They studied the soil by using* (A) plants (B) a telescope (C) roots **(D) a microscope**
Related Evidence Facts:	[1] Studying a soil sample means studying the small microorganisms in that soil. [2] Magnifying makes seeing microorganisms easier through using a microscope.
Reasoning Chain:	Question $\xrightarrow{\texttt{soil}}$ [1] $\xrightarrow{\texttt{microorganism}}$ [2] $\xrightarrow{\texttt{microscope}}$ (D)

(e) Case Study 5

Question:	*Birds carrying away fruit helps the tree* (A) grow (B) fertilize **(C) reproduce** (D) conquer
Related Evidence Facts:	[1] Birds are a vehicle for spreading the seeds of a plant. [2] Ex2: plants reproduce with seeds.
Reasoning Chain:	Question $\xrightarrow{\texttt{bird}}$ [1] $\xrightarrow{\texttt{plant}}$ [2] $\xrightarrow{\texttt{reproduce-01}}$ (C)

(f) Case Study 6

Question:	*The salamander could eat a large amounts of what?* (A) fettuccine (B) waxy leaves from certain plants (C) dead carcass meat from livestock **(D) six legged winged organisms**
Related Evidence Facts:	[1] A salamander eats insects. [2] Insects have three parts to their bodies, wings, two feelers, and six legs.
Reasoning Chain:	Question $\xrightarrow{\texttt{salamander}}$ [1] $\xrightarrow{\texttt{insect}}$ [2] $\xrightarrow{\texttt{wing}}$ (D)

Table 7: More case studies in addition to Table 6