RPA: Reasoning Path Augmentation in Iterative Retrieving for Multi-Hop QA

Ziyi Cao, Bingquan Liu, Shaobo Li

Harbin Institute of Technology
zyc@stu.hit.edu.cn, liubq@hit.edu.cn, shli@insun.hit.edu.cn

Abstract

Multi-hop questions are associated with a series of justifications, and one needs to obtain the answers by following the reasoning path (RP) that orders the justifications adequately. So reasoning path retrieval becomes a critical preliminary stage for multi-hop Question Answering (QA). Within the RP, two fundamental challenges emerge for better performance: (i) what the order of the justifications in the RP should be, and (ii) what if the wrong justification has been in the path. In this paper, we propose **R**easoning **P**ath **A**ugmentation (RPA), which uses reasoning path reordering and augmentation to handle the above two challenges, respectively. Reasoning path reordering restructures the reasoning by targeting the easier justification first but difficult one later, in which the difficulty is determined by the overlap between query and justifications since the higher overlap means more lexical relevance and easier searchable. Reasoning path augmentation automatically generates artificial RPs, in which the distracted justifications are inserted to aid the model recover from the wrong justification. We build RPA with a naive pre-trained model and evaluate RPA on the QASC and MultiRC datasets. The evaluation results demonstrate that RPA outperforms previously published reasoning path retrieval methods, showing the effectiveness of the proposed methods. Moreover, we present detailed experiments on how the orders of justifications and the percent of augmented paths affect the question-answering performance, revealing the importance of polishing RPs and the necessity of augmentation.

Introduction

Multi-hop QA is the Question Answering (QA) task taking account of information from multiple justifications and reasoning the final answer (Yang et al. 2018; Khashabi et al. 2018a; Khot et al. 2020). Especially, the retrieval for multi-hop QA is distinct from single-hop QA since multiple justifications are required to support the answering. The requirement of multiple pieces further demands multiple retrieving steps to collect them. Multi-step retrieval can be conventionally carried out by iterative single-step retrieving (Asai et al. 2020; Das et al. 2019; Feldman and El-Yaniv 2019), where the current retrieving step draws on the previous searching results, forming the reasoning path (RP). The iterative retrieving process works fine when all sentences available in the searched RP are justifications. Nevertheless, it is unrealistic to guarantee that. If one retrieval step is mistaken leading to a false justification in RP, it will result in significant errors in the later retrieving steps since it could mislead all the following steps. The above phenomenon can be referred to as the cascading failure in retrieval, which hinders an iterative process from finally success.

Q : Who didn't stay in **Zurich** after **Albert** and Maric **separated**?
(G) **Einstein**

Reasoning Path (*RP*) :

$\hat{f}_1$: In May 1904, the **couple**'s first **son**, Hans **Albert Einstein**, was **born** in Bern, Switzerland.

$\hat{f}_2$: Their second **son**, Eduard, was **born** in **Zurich** in July 1910.

$\hat{f}_3$: In 1914, the **couple separated**; **Einstein** moved to Berlin and his wife remained in **Zurich** with their **sons**.

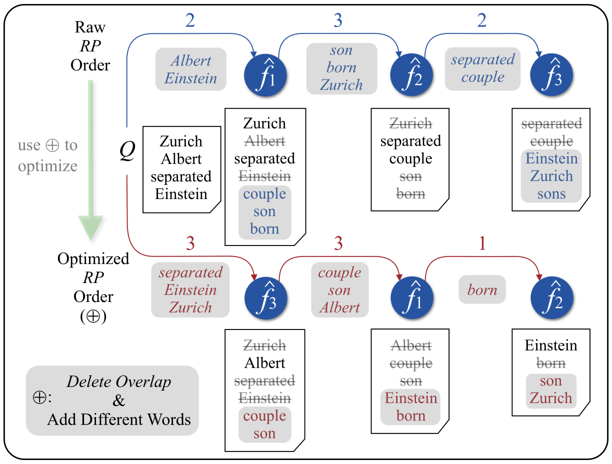


Figure 1: A sample in MultiRC is supported by three justifications with another justifications order optimized by the operation ⊕ (the exclusive-or between two word-sets, $i.e.$, delete overlap and add different words). All words useful for overlap are bold, and the overlap number is noted on the line, $e.g.$, in the raw RP order, the number (2) on the line from Q to $\hat{f}_1$ means the overlap number between them, and the number (2) from $\hat{f}_2$ to $\hat{f}_3$, whose root is Q via $\hat{f}_1$, is denoted as the overlap between $Q \oplus \hat{f}_1 \oplus \hat{f}_2$ with $\hat{f}_3$.

To mitigate the hindrance, an obvious way is to train the information retriever (IR) to continue searching for the correct target despite the emergence of false justifications in RPs. It links out to several unavoidable challenges including, what is the better path to reason in justifications, as well as how to construct the training samples to help the IR recover from the information with a false message, to name a few. The challenges mentioned are supposed to be dealt with sequentially, $i.e.$, make sure the RPs are more suitable for retrieving and then build samples based on them.

In the challenge of the RP, we presume that the later the error occurs, the less loss. To simplify the situation, we define three ground-truth justifications $\{\hat{f}_1, \hat{f}_2, \hat{f}_3\}$ to search with query Q[1], which at least needs three retrieval steps with one error unavoidable in the steps. If the error appears at the first step, it can result in a negative effect for all of the following retrieving. On the other hand, if the error's position is at the third step, the other two before are with no influence. The performance of the retrieval in the first step is decided by the ability of the IR, resulting from both the difficulty of the searching target at the first step and the basic capability of the IR. The basic capability is determined by the pre-trained model while the degree of first-step retrieving difficulty seems to be associated with the first sample. If the sample for the first step is easier to understand, such as using the overlap number of words, an obvious method, to get the most relevant justification, the model is supposed to be easier to train for higher scores. As shown in Figure 1, the raw justifications order in RP (raw RP order) is provided by the dataset, whose first target is $\hat{f}_1$ but $\hat{f}_3$ in optimized RP order. In the aspect of the overlap, $\hat{f}_3$ should be easier to be retrieved than $\hat{f}_1$ with Q for its higher overlap.

With $\hat{f}_3$ in the first position, we concentrate on the new entities introduced in $\hat{f}_3$ that are absent from Q and the old unsearched entities that are in Q but not in $\hat{f}_3$ (Khot et al. 2020), collectively referred to as $Q \oplus \hat{f}_3$ ($e.g.$, "*couple*", "*son*", "*Albert*" and rest irrelevant words). We use the overlap with Q to get $\hat{f}_3$ and, similarly, get $\hat{f}_1$ based on the overlap with $Q \oplus \hat{f}_3$. In such a way, we rebuild the order of the RP as $\{\hat{f}_3, \hat{f}_1, \hat{f}_2\}$ finally.

For building samples, the challenge to handle is how to generate training data relying on the RPs. Within the sample shown in Figure 1, the original training data can be generated such as, using Q to predict $\hat{f}_3$ and using Q and $\hat{f}_3$ to predict $\hat{f}_1$, just for instance. However, it has no idea to deal with the cascading failure, which indicates the existence of the incorrect sentence in the RPs. A most straightforward way for the model to allow its existence is to build the artificial RPs that contain the false justifications, $e.g.$, using Q and f to predict $\hat{f}_3$ ($f \notin \{\hat{f}_1, \hat{f}_2, \hat{f}_3\}$) (Asai et al. 2020). We extract the f from the approximate nearest neighbor (ANN) (Xiong et al. 2021) searched from the specific knowledge base (KB) with Q. In this paper, we augment the training data by mixing different percentages of artificial data into the original data, showing the detailed improvement of the mixture of augmentation.

Our contributions. To be more specific, this paper introduces the Reasoning Path Augmentation (RPA), a method combining the reasoning path reordering and augmentation. For each RP, we manually reorder it and rebuild different ratios of artificial paths for training. Additionally, aiming to highlight the improvement of our methods, we use naive RoBERTa pre-trained model as encoder and inner product as similarity method in training to prevent the influence of specially designed structure for multi-step. Experiments conclude that our proposed approach outperforms the existing retrieval methods on the specific datasets.

Our prime contributions are as follows:

- We propose to reorder the justifications sequence in the reasoning path with the overlap between query and justifications, placing the hard-to-fetch gold justifications behind that reduces the difficulty of the searching ahead.
- We propose to automatically insert the incorrect retrieved sentence of the approximate nearest neighbor as the confusing justification into the reasoning path, contributing different ratios of artificial training data.
- Evaluated on QASC (Khot et al. 2020) and MultiRC (Khashabi et al. 2018a), the proposed approach significantly outperforms previously published reasoning path retrieval methods and presents a further experimental analysis.

Related Works

Reasoning path. Within the task of multi-hop QA, the question is answered by gathering information from multiple justifications (Khashabi et al. 2018a) connected with the question or each other. In QASC, the baseline formed with only two iterations uses the $Q \oplus f_1$ as the connection to identify the second justification by Lucene (f_1 is the sentence retrieved in the first step). However, this designed method for dataset-specific is powerless against other conditions of data. Avoiding this, Yadav et al. (2020) propose AIR, using the matrix of cosine similarity scores between the question and each word in the KB, iteratively removes the words retrieved and searches again, in which the RP is constructed unsupervised on both QASC and MultiRC. The similarity scores between tokens are effective in removing invalid information while keeping valid keywords, nevertheless, ignoring the contextual representation of the question. Additionally, Yadav et al. (2021) propose the JointRR uses the same method of similarity score in AIR as the filter but with a RoBERTa re-ranker, presenting a better performance of retrieval. The answer classifier in AIR and JointRR is the same RoBERTa. The difference between AIR and our approach is that we use the naive pre-trained model and similarity method of inner product, simplifying the retriever structure, to concentrate on the reasoning path augmentation.

Cascading failure. The cascading failure in retrieval can result in a significant drop in performance with iteration despite a high IR score, which is ignored in both AIR and JointRR. To handle this challenge, Asai et al. (2020) propose the PathRetriever, inserting the artificial RPs which are added

[1] Typical IR approaches for QA retrieve justifications using question + answer (q + a) as their IR query (Clark et al. 2016; Khot, Sabharwal, and Clark 2017; Khashabi et al. 2018b).

to the high TF-IDF score paragraphs at the first position. The augmentation of the RPs in PathRetriever can be useful in searching at the second position but helpless in the other positions reasoning, while the paragraph added is sorted by TF-IDF but not the model, which reduces the difficulty of confusion. To enhance the stabilization, we select a more confusing sentence from the model-retrieved results of the question and build augmentation automatically while training, paralleling running at the same time. The justification selected is various with the searching based on either Q or Q and $\hat{f}_i$ ($\hat{f}_i$ is in the ground-truth justifications), and dynamically inserted in any position, forming artificial RPs.

Justifications order. Only supervised methods are associated with the order of justifications in RP (Asai et al. 2020; Li et al. 2021), while the unsupervised methods ignore it (Yadav et al. 2020, 2021). PathRetriever adaptively scores each RP in the graph constructed with the Wikipedia hyperlinks and document structures to model the relationships. However, to visualize the detailed performance difference of the changing in the justifications order, we give up the specially designed structure for multi-hop yet adopt the naive pre-trained model for better universality.

Approach

Overview

Multi-hop QA consists of two components: (i) a retrieval component and (ii) an answer component (Chen et al. 2017; Yadav et al. 2020). In this paper, our proposed method concentrates on the retrieval task while keeping the answer component as standard (Yadav et al. 2020).

Task definition. Our retrieval task is defined as Iterative Multi-hop IR for QA: (i) Retrieve N_1 facts F_1 as justifications from facts corpus F_C based on the query $Q = q+a$; (ii) For each $f_{M-1} \in F_{M-1}$ ($M > 1, M \in \mathbb{N}^+$), iteratively retrieve N_M justifications $F_M = \{f_1, \cdots, f_{N_M}\}_{N_M}$ based on Q and $\{f_1, \cdots, f_{M-1}\}$; (iii) Select top K unique facts from the reasoning paths $\{f_1, \cdots, f_M\}$ sorted by the sum of their individual retrieval score. Each supervised training sample, *i.e.*, one idealized reasoning step, for the IR is defined as

$$\hat{f}_M = IR(Q, \{f_1, \cdots, f_{M-1}\}),$$

in which $\hat{f}_M$ means the training target, hence $\{\hat{f}_1, \hat{f}_2, \cdots\}$ represents all ground-truth justifications in a RP. $\{N_i\}_M$ is used to indicate that the RPs are predicted in M steps while N_i justifications are obtained at the i^{th} step ($i \leq M$).

Our method. Based on this intuition, each retrieval query Q in our proposed method is run against RPA, whose basic structure is ANCE (Xiong et al. 2021), a single-hop retriever, iteratively retrieves justifications from dataset-specific KBs. For example, in MultiRC, we use all sentences in the paragraph as F_C for a given query. In QASC, which has a large KB of 17 million sentences, 80 sentences fetched by Lucene in Heuristic+IR method[2] (Khot et al. 2020) from the provided QASC KB are viewed as F_C for each candidate answer to reducing noise.

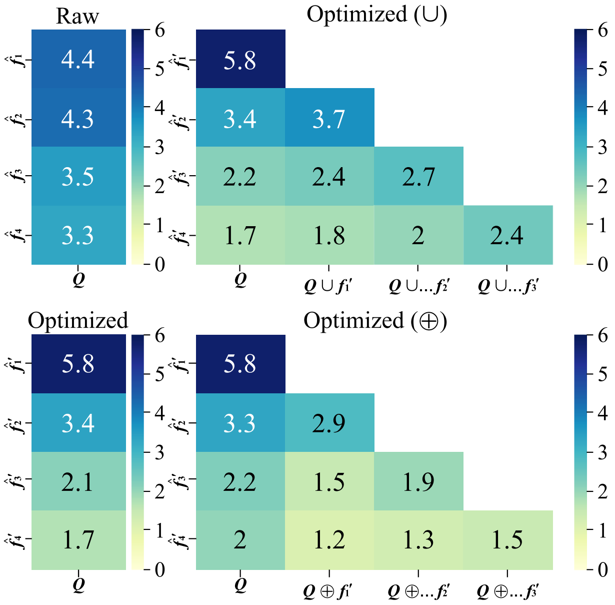


Figure 2: The average number of overlap in MultiRC.

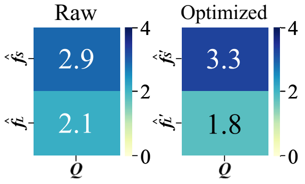


Figure 3: The average number of overlap in QASC.

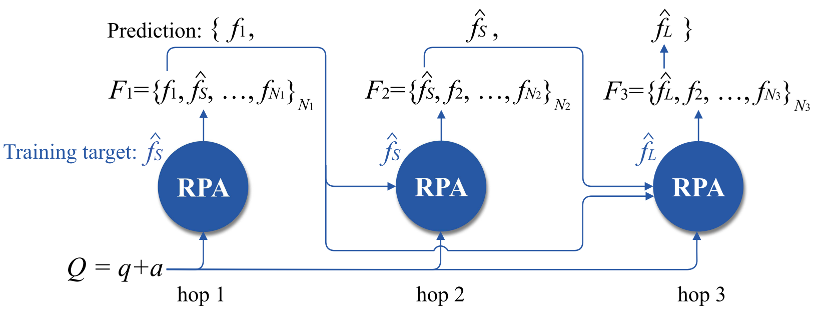
With the goal of easier reasoning and preventing cascading failures, we present two methods of reasoning path augmentation in the multi-hop retrieval: (i) reorder the sequence of justifications in the RP; and (ii) automatically create artificial paths as augmented training samples. The details are discussed in the following two sub-sections.

Justifications Reordering

Every training item of RPA is built following the justifications in RP. Raw RP order follows the justifications sequence provided by the dataset, while in the optimized RP order, the more overlapping preprocessed words[3] of $\hat{f}_i$ with Q ($i = 1$) or $Q \oplus \hat{f}_1 \oplus \cdots \hat{f}_{i-1}$ ($i > 1$) , the more forward the position since the overlap can denote simple lexical relationships between sentences. With more relevant sentences selected at the steps ahead, which is easier for IR to fetch at each step, reordering by overlap is supposed to obtain better entire performance than the raw order. For example, in MultiRC, whose overlap is shown in Figure 2, each RP of development samples consists of 2 to 4 justifications $\{\hat{f}_1, \hat{f}_2, \cdots\}$, and their order optimization is shown in Figure 1. In QASC, ev-

[2] In QASC fetched 80 sentences, Recall of at least one justification found is 81.8, and Recall of both found is 61.3, which determines the upper-bound of the retrieval performance of RPA.

[3] We use the set-intersection between two stemmed, nonstopword word-sets (Khot et al. 2020) in Q and $\hat{f}_i$ to identify the overlap in QASC and MultiRC.



Prediction: $\{ f_1,$
$\hat{f}_S ,$
$\hat{f}_L \}$
$F_1=\{f_1, \hat{f}_S, \ldots, f_{N_1}\}_{N_1}$
$F_2=\{\hat{f}_S, f_2, \ldots, f_{N_2}\}_{N_2}$
$F_3=\{\hat{f}_L, f_2, \ldots, f_{N_3}\}_{N_3}$
Training target: $\hat{f}_S$
$\hat{f}_S$
$\hat{f}_L$
RPA
RPA
RPA
$Q = q+a$
hop 1
hop 2
hop 3

Figure 4: RPA is trained with the target following RP and the target in hop 2 is still $\hat{f}_S$, for example, which results in the number of training steps being expanded to 3 and the targets are $\{\hat{f}_S, \hat{f}_S, \hat{f}_L\}$. All outputs are identified as the prediction $\{f_1, \hat{f}_S, \hat{f}_L\}$.

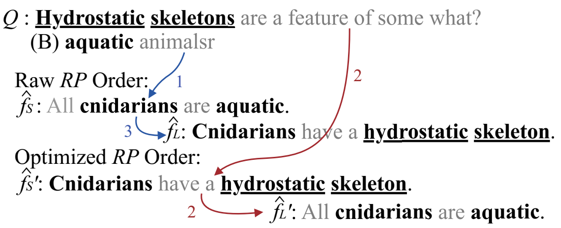

Q : **<u>Hydrostatic</u> <u>skeletons</u>** are a feature of some what?
(B) **aquatic** animalsr
Raw *RP* Order:
$\hat{f}_S$: All **cnidarians** are **aquatic**.
$\hat{f}_L$: **Cnidarians** have a **<u>hydrostatic</u> <u>skeleton</u>**.
Optimized *RP* Order:
$\hat{f}_S{}'$: **Cnidarians** have a **<u>hydrostatic</u> <u>skeleton</u>**.
$\hat{f}_L{}'$: All **cnidarians** are **aquatic**.

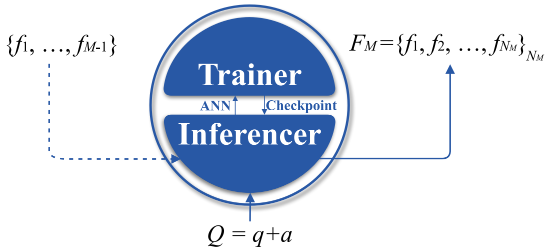
Figure 5: Raw RP Order $v.s.$ Optimized RP Order in QASC

ery RP is always associated with justifications $\{\hat{f}_S, \hat{f}_L\}$ such that the optimization algorithm of order is simple, where we present Figure 3 for its overlap and Figure 5 for the optimization.

Figure 2 also demonstrates the overlap of the other two types of the order optimization algorithm, $i.e.$, Q and $Q \cup \hat{f}_1 \cup \cdots \hat{f}_{i-1}$[4] with $\hat{f}_i$ on MultiRC (all three optimization algorithms are the same on QASC). The results of the two types are shown in Figure 7b, presenting a slight enhancement on the raw order yet a gap to the algorithm $\oplus$.

Artificial Reasoning Path

Original training data produced relying on the RPs is like $\hat{f}_M = IR(Q, \{f_1, \cdots, f_{M-1}\})$. Specifically, in QASC, original data includes $\hat{f}_S = IR(Q)$, $\hat{f}_S = IR(Q, \{\hat{f}_L\})$, $\hat{f}_L = IR(Q, \{\hat{f}_S\})$[5]. To reduce the effect of the cascading failure, we build artificial samples which contain searching failure while hopping. Based on RPA, we manually adopt the **top-1 non-ground-truth** sentence fetched (such as f_1 in hop 1 in Figure 4) as the distracted justification to be inserted into RP, which produces augmented data automatically, $e.g.$, $\hat{f}_S = IR(Q, \{f_1\})$, $\hat{f}_L = IR(Q, \{f_1, \hat{f}_S\})$, $\hat{f}_S = IR(Q, \{f_1, \hat{f}_L\})$, $\hat{f}_S = IR(Q, \{\hat{f}_L, f_2\})$, $\hat{f}_L = IR(Q, \{\hat{f}_S, f_3\})$ (f_1, f_2, f_3 are the top-1 non-correct prediction of $IR(Q)$, $IR(Q, \{\hat{f}_L\})$, $IR(Q, \{\hat{f}_S\})$, respectively.). The scale of entire artificial RPs is 5/3 times the original data (length of RP is 2) since the distracted justification can be inserted in any position of RP, augmenting the dataset for training and empowering the IR the ability to recover from the searching failure. In MultiRC, artificial RPs are generated similarly. Training data for RPA involves the total original data and various percent ($p\%$) of artificial data.



$\{f_1, \ldots, f_{M-1}\}$
$F_M=\{f_1, f_2, \ldots, f_{N_M}\}_{N_M}$
Trainer
ANN
Checkpoint
Inferencer
$Q = q+a$

Figure 6: RPA basic structure: Trainer and Inferencer. Parallel iterative execution: (i) Trainer runs with the latest ANN and saves checkpoint; (ii) Inferencer loads the latest checkpoint and generates ANN.

Experiment

Setup

Pipeline. The whole procedure follows a coarse-to-fine pipeline that contains three stages:

- Preliminary retrieval: 80 sentences are used as F_C to be retrieved for each answer in QASC; All content of the paragraph is considered as F_C in MultiRC.
- Justifications retrieval: Retrieve the justifications by RPA on F_C iteratively.
- Answer classifier: The answer a concatenated with q and retrieved justifications is classified on RoBERTa, the same as AIR (Yadav et al. 2020).

Details. Hyperparameters of RPA, whose basic structure is shown in Figure 6, are from the Trainer that is with saving steps of 2 and epoch of 500×saving steps, fine-tuned from RoBERTa-Large (Liu et al. 2019; Wolf et al. 2020; Xiong et al. 2021). More specifically, we trained with batch size of 20 in QASC and 4 in MultiRC[6], chunk size of 50, and the

[4] $\cup$ means the set-union of two word-sets.

[5] q and a are concatenated in the way of “Query: q Choice: a” to get Q, similarly, Q and $\{f_1, \cdots, f_{M-1}\}$ are integrated into “Q Fact: $f_1 \cdots$ Fact: f_{M-1}”, and f in F_C is refactored as “Fact: f” before being embedded (Liu et al. 2021).

[6] Epoch in MultiRC is 200×saving steps for early stopping due to its smaller batch size and smaller F_C.

#	Method (RP order type, $p\%$ of artificial data)	Prediction	$\{N_i\}_M$	Accuracy	Recall@10	Recall@10
		Hops			*both*	*at least one*
		M			*found*	*found*
	DEVELOPMENT SET					
	Baselines					
1	Naive Lucene BM25 (Yadav et al. 2020)	1	-	35.6	17.2	68.1
2	Naive Lucene BM25 (Yadav et al. 2020)	2	-	36.3	27.8	65.7
3	Heuristics+IR (Khot et al. 2020)	2	{20,4}	32.4	41.6	64.6
4	ESIM Q2Choice (Khot et al. 2020)	-	-	32.4	41.6	64.6
	Previous Work					
5	BERT-LC (Khot et al. 2020)	1	-	59.8	11.7	54.7
6	BERT-LC (Khot et al. 2020)	2	-	71.0	41.6	64.4
7	BERT-LC[WM]* (Khot et al. 2020)	2	-	78.0	41.6	64.4
8	AIR top chain + RoEBRTa (Yadav et al. 2020)	2	{1,1}	76.2	-	-
9	AIR (parallel=5) + RoBERTa (Yadav et al. 2020)	2	{5,1}	81.4	44.8	68.6
10	SingleRR + RoBERTa (Yadav et al. 2021)	2	-	79.7	44.4	69.6
11	JointRR + RoBERTa (Yadav et al. 2021)	2	-	81.7	45.3	69.4
	RPA + RoBERTa (M=2 or 3)					
12	RPA + RoBERTa (raw order, 0)	2	{1,9}	83.9	48.0	76.6
13	RPA + RoBERTa (optimized order, 0)	3	{1,1,8}	85.1	49.4	77.1
14	RPA + RoBERTa (optimized order, 90)	3	{1,1,8}	85.0	**55.2**	79.5
15	RPA + RoBERTa (optimized order, 100)	3	{1,1,8}	**86.0**	54.9	**79.8**
	TEST SET					
16	BERT-LC (Khot et al. 2020)	2	-	68.5		
17	BERT-LC[WM]* (Khot et al. 2020)	2	-	73.2		
18	JointRR + RoBERTa (Yadav et al. 2021)	2	-	78.0		
19	AIR (parallel=5) + RoBERTa (Yadav et al. 2020)	2	{5,1}	81.4		
20	RPA + RoBERTa (optimized order, 90)	3	{1,1,8}	**81.5**		

Table 1: Performance of QA and justifications retrieval (top $K = 10$) on QASC.

optimizer of Lamb (You et al. 2020), whose learning rate is 5e-6. For training, in QASC, we uniformly sampled 5 negatives from ANN top 100, and in MultiRC, sampled 2 negatives from all sentences (the number of f in F_C varies from 6 to 20 in training data). In the answer classifier, we used batch size 2, maximum sequence length 256 for QASC[7].

RPA is trained to ignore retrieval stop during hopping so that we present a simple yet reasonable approach to stop the iteration on MultiRC: (i) word set of Q removes the words that are in Q but not in F_C (unsearchable words) and (ii) also removes the words in retrieved justifications, (iii) if there are remaining words in the Q word set, then continue to retrieve, else stop the iteration (maximum iteration hops are 4).

Evaluation

We evaluated our method on two datasets:

Question Answering via Sentence Composition (QASC), a large KB-based multiple-choice QA task[8] (Khot et al. 2020). Each sample consists of a question with 8 answer candidates, out of which 4 candidates are hard adversarial choices. Every question is annotated with a fixed set of two justifications $\{\hat{f}_S, \hat{f}_L\}$ as raw RP order for answering the question.

Multi-Sentence Reading Comprehension (MultiRC), a reading comprehension dataset consists of multiple choices QA (Khashabi et al. 2018a). In the development set, every question with 2-to-14 answer candidates is supported with a paragraph, which contains 2-to-4 justifications. The dataset we use is the original MultiRC[9] including the ground-truth justifications, but not the version on SuperGLUE (Wang et al. 2019). The original MultiRC contains the training, development, and hidden test set, out of which the training and development set is used in the paper.

We report question-answering performance as well as justifications collection performance in Table 1 for QASC and Table 2 for MultiRC[10].

Results

Justifications collection. In QASC, we report Recall@10 similar to (Khot et al. 2020; Yadav et al. 2020). *both found* reports the recall scores when both the gold justifications are found in the top $K = 10$ ranked sentences and similarly, *at*

[7] Hyperparameters of answer selector are the same with AIR's RoBERTa unless otherwise indicated and all random seeds are 0.

[8] https://leaderboard.allenai.org/qasc/submissions/public

[9] https://cogcomp.seas.upenn.edu/multirc/

[10] All results except the baselines should include the performance about the justifications collection as much as possible.

#	Method (RP order type, $p\%$ of artificial data)	F1$_m$	F1$_a$	EM0	Justifications collection		
					P	R	F1
	DEVELOPMENT SET						
	Baselines						
1	IR(paragraphs) (Khashabi et al. 2018a)	64.3	60.0	1.4	-	-	-
2	SurfaceLR (Khashabi et al. 2018a)	66.5	63.2	11.8	-	-	-
3	Entailment baseline (Trivedi et al. 2019)	51.3	50.4	-	-	-	-
	Previous work						
4	RS* (Sun et al. 2019)	73.1	70.5	21.8	-	-	60.8
5	BERT + BM25 (Yadav et al. 2019)	71.1	67.4	23.1	43.8	61.2	51.0
6	BERT + AutoROCC (Yadav et al. 2019)	72.9	69.6	24.7	48.2	68.2	56.4
7	Entire passage + RoBERTa (Yadav et al. 2020)	73.9	71.7	28.7	17.4	100.0	29.6
8	RoBERTa-retriever(All passages) + RoBERTa (Yadav et al. 2020)	70.5	68.0	24.9	63.4	61.1	62.3
9	AIR top chain + RoBERTa (Yadav et al. 2020)	74.7	72.3	29.3	66.2	63.1	64.2
10	AIR (parallel = 5) + RoBERTa (Yadav et al. 2020)	77.2	75.1	33.0	28.6	84.1	44.9
11	JointRR + RoBERTa (Yadav et al. 2021)	75.2	72.7	28.2	65.4	69.9	67.6
12	JointRR (± 1 neighboring sentence) + RoBERTa (Yadav et al. 2021)	77.0	74.5	32.9	**65.4**	69.9	**67.6**
	RPA + RoBERTa						
13	RPA + RoBERTa (raw order, 0)	77.2	74.7	31.2	63.1	71.1	66.8
14	RPA + RoBERTa (raw order, 10)	76.4	74.5	30.9	62.6	71.7	66.9
15	RPA + RoBERTa (optimized order $\oplus$, 0)	77.1	74.6	31.4	63.8	71.1	67.3
16	RPA + RoBERTa (optimized order $\oplus$, 10)	77.1	74.6	31.3	64.0	**72.0**	**67.6**
17	RPA + RoBERTa (optimized order $\oplus$, 100)	76.7	74.3	30.2	62.6	71.9	66.9
18	RPA (parallel = 5) + RoBERTa (optimized order $\oplus$, 10)	**78.3**	**75.9**	**33.1**	37.5	86.1	52.3
	Ceiling systems with gold justifications						
19	Oracle knowledge + RoBERTa (Yadav et al. 2021)	81.4	80	39	100.0	100.0	100.0
20	Human	86.4	83.8	56.6	-	-	-

Table 2: Results on the dataset of MultiRC.

least one found reports the recall scores when either one or both the gold justifications are found in the top 10 ranked sentences, as shown in Table 1. In Table 2 for MultiRC, we report Precision (P), Recall (R), and F1 to evaluate the performance of the justifications retrieval. Evaluated on QASC and MultiRC, RPA outperforms all the aforementioned baseline models and other published retrieving methods.

To be fairly compared with AIR, which is the state-of-the-art published method in reasoning path prediction on QASC[11], RPA uses the same initial search space (*i.e.* top-80 sentences based on Lucene scores) and answer classifier (*i.e.* RoBERTa-Large) with AIR. RPA ({1,9}, raw order, 0) outperforms the AIR by 3.2% and 8% on top-10 justifications collection metrics respectively. Moreover, the result of RPA with 100% augmented data increases by 5.5% and 2.7% in the same condition with optimized order and $\{N_i\}_M$ {1,1,8}. In MultiRC, RPA also yields satisfactory performance. Considering the parallel of 5 in RPs in AIR, we adopt the same process in RPA[12] and get better results.

Answer classification. The answer classifier benefits from the improvements in justifications retrieving (Li et al. 2021) so that RPA also demonstrates considerably better results while classifying the answer on both QASC[13], which is classified as multi-choice QA (MCQA) (Wolf et al. 2020), and MultiRC, which is evaluated with the correctness of each answer individually.

Analysis

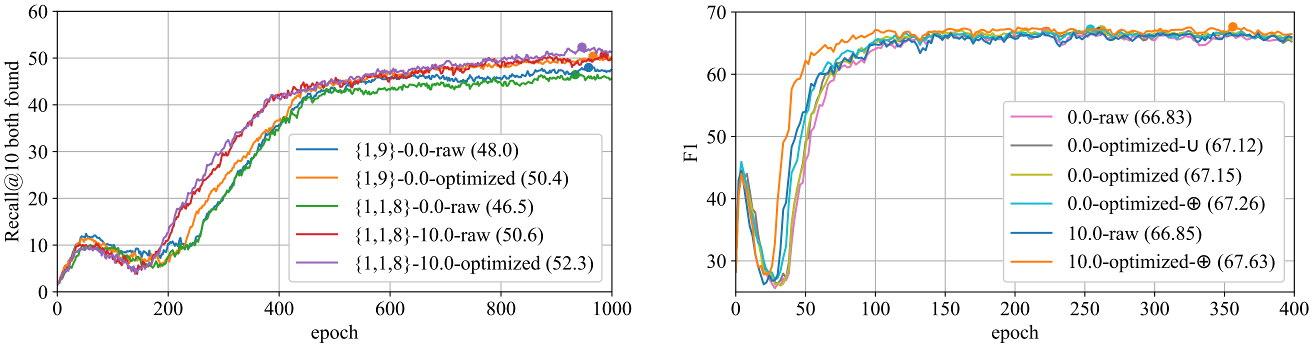
Overlap and justifications order. As shown in Figures 2 and 3, the minor adjustment to the justifications order in RP increases the polarization of the average number of overlapping words. The adjustment reduces the difficulty in lexical searching step by step, further slowing down the hardness in semantic retrievals. In this way, the target in each retrieval step can be more accessible for a pre-trained model and further modifies the distribution of the model's embedding space, resulting in a better comprehensive performance in the reasoning path retrieval.

Justification prediction in hops. In QASC, to visualize the predicting variation, we aggregate the Recall@1 of $\hat{f}_S$ and $\hat{f}_L$ with optimized RP order and $\{N_i\}_M$ of {1,1,1} into Figure 8. Since the augmented data is mixed in training,

[11] Other methods on QASC outperforming AIR are published with no justifications selection results.

[12] The 5 parallel retrievals of RPA with stop method in MultiRC is same with $\{N_i\}_M$ {5,1,1,1}.

[13] The upper-bound decided by Lucene is unavailable on QASC test set. If the test-set upper bound is lower, there will be less lifting space for our method, resulting in the less improvement in #20 of Table 1.



(a) QASC ($\{N_i\}_M$-$p\%$-order (maximum))

(b) MultiRC with stop method ($p\%$-order-operation (maximum))

Figure 7: Evaluation of datasets with each improvement method or both in justifications collection. The maximum of each line is pointed in the line and noted in the figure legend.

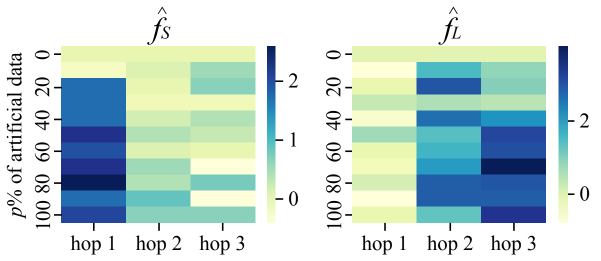


Figure 8: Recall increment of $\hat{f}_S$ and $\hat{f}_L$ varies with $p\%$ in each hop (3 hops, optimized RP order, $\{N_i\}_M$ of $\{1,1,1\}$). The $\{1,1,1\}$ is extracted from $\{1,1,8\}$, and all results has subtracted the result of $p = 0$.

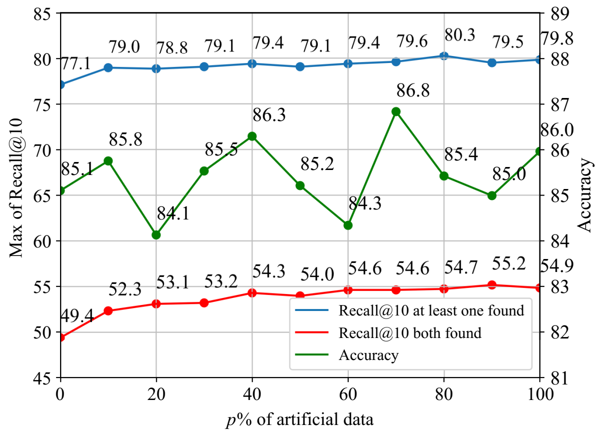


Figure 9: Maximum Recall@10 and Accuracy on QASC vary with $p\%$ (optimized RP order, Prediction Hops of 3, $\{N_i\}_M$ of $\{1,1,8\}$).

the results of $\hat{f}_S$ and $\hat{f}_L$ predicted in hops 2 and 3 improve with the increasing $p\%$ as expected. Surprisingly, the performance of $\hat{f}_S$ in hop 1 also improves. However, there is no existence of $\hat{f}_S = IR(Q)$ in artificial data such that the training of $\hat{f}_S = IR(Q)$ would not change with increasing $p\%$, indicating that the increasing percentage of the augmented data empowers the entire ability of retrieval. In MultiRC, despite the influence of the overfitting and the non-fixed prediction length, the artificial data can also make a slight enhancement with a smaller $p\%$.

Ablation study. As shown in Figure 7a, RPA with optimized RP order on QASC always outperforms it with the raw order in not only the model convergence speed but also the maximum of the performance under the same $\{N_i\}_M$ and $p\%$ conditions, which indicates the great potential of the RP reordering. Additionally, the 10% augmentation of RPs contributes a similar enhancement from {1,1,8}-0.0-raw-order to {1,1,8}-10.0-raw-order, and the superposition of the two methods is to present further improvement than a single one. The same analysis can also be obtained on MultiRC with $p\%$ from 0 to 10 and the order from the raw order to the optimized ($\oplus$) order as shown in Figure 7b. Further ablation experiments are conducted in Figure 9 to corroborate the necessity of the artificial RPs since a mere 10% of artificial data can provide a significant boost of 3% while the boost becomes slower as the $p\%$.

Conclusion

In this paper, we propose the Reasoning Path Augmentation (RPA) to reorder and augment the reasoning paths with RoBERTa. The order optimization algorithm of the justifications reduces the difficulty of sequential iterative retrievals by the overlap words between the query and justifications, while augmented data produced further enhances the performance in multi-hop retrieval, which outperforms the published retrieval methods on QASC and MultiRC. Additionally, the analysis of the reasoning paths further demonstrates the necessity of justification order optimization and reasoning path augmentation.

References

Asai, A.; Hashimoto, K.; Hajishirzi, H.; Socher, R.; and Xiong, C. 2020. Learning to Retrieve Reasoning Paths over Wikipedia Graph for Question Answering. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Chen, D.; Fisch, A.; Weston, J.; and Bordes, A. 2017. Reading Wikipedia to Answer Open-Domain Questions. In Barzilay, R.; and Kan, M., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 1870–1879. Association for Computational Linguistics.

Clark, P.; Etzioni, O.; Khot, T.; Sabharwal, A.; Tafjord, O.; Turney, P. D.; and Khashabi, D. 2016. Combining Retrieval, Statistics, and Inference to Answer Elementary Science Questions. In Schuurmans, D.; and Wellman, M. P., eds., *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, 2580–2586. AAAI Press.

Das, R.; Dhuliawala, S.; Zaheer, M.; and McCallum, A. 2019. Multi-step Retriever-Reader Interaction for Scalable Open-domain Question Answering. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Feldman, Y.; and El-Yaniv, R. 2019. Multi-Hop Paragraph Retrieval for Open-Domain Question Answering. In Korhonen, A.; Traum, D. R.; and Màrquez, L., eds., *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2296–2309. Association for Computational Linguistics.

Khashabi, D.; Chaturvedi, S.; Roth, M.; Upadhyay, S.; and Roth, D. 2018a. Looking Beyond the Surface: A Challenge Set for Reading Comprehension over Multiple Sentences. In Walker, M. A.; Ji, H.; and Stent, A., eds., *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, 252–262. Association for Computational Linguistics.

Khashabi, D.; Khot, T.; Sabharwal, A.; and Roth, D. 2018b. Question Answering as Global Reasoning Over Semantic Abstractions. In McIlraith, S. A.; and Weinberger, K. Q., eds., *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 1905–1914. AAAI Press.

Khot, T.; Clark, P.; Guerquin, M.; Jansen, P.; and Sabharwal, A. 2020. QASC: A Dataset for Question Answering via Sentence Composition. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, 8082–8090. AAAI Press.

Khot, T.; Sabharwal, A.; and Clark, P. 2017. Answering Complex Questions Using Open Information Extraction. In Barzilay, R.; and Kan, M., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 2: Short Papers*, 311–316. Association for Computational Linguistics.

Li, S.; Li, X.; Shang, L.; Jiang, X.; Liu, Q.; Sun, C.; Ji, Z.; and Liu, B. 2021. HopRetriever: Retrieve Hops over Wikipedia to Answer Complex Questions. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 13279–13287. AAAI Press.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2021. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. *CoRR*, abs/2107.13586.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR*, abs/1907.11692.

Sun, K.; Yu, D.; Yu, D.; and Cardie, C. 2019. Improving Machine Reading Comprehension with General Reading Strategies. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 2633–2643. Association for Computational Linguistics.

Trivedi, H.; Kwon, H.; Khot, T.; Sabharwal, A.; and Balasubramanian, N. 2019. Repurposing Entailment for Multi-Hop Question Answering Tasks. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 2948–2958. Association for Computational Linguistics.

Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; and Bowman, S. R. 2019. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 3261–3275.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; Davison, J.; Shleifer, S.; von Platen, P.; Ma, C.; Jernite, Y.; Plu, J.; Xu, C.; Scao, T. L.; Gugger, S.; Drame, M.; Lhoest, Q.; and Rush, A. M. 2020. Transformers: State-of-the-Art Natural Language Processing. In Liu, Q.; and Schlangen, D., eds., *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, EMNLP 2020 - Demos, Online, November 16-20, 2020*, 38–45. Association for Computational Linguistics.

Xiong, L.; Xiong, C.; Li, Y.; Tang, K.; Liu, J.; Bennett, P. N.; Ahmed, J.; and Overwijk, A. 2021. Approximate Nearest Neighbor Negative Contrastive Learning for Dense Text Retrieval. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Yadav, V.; et al. 2019. Quick and (not so) Dirty: Unsupervised Selection of Justification Sentences for Multi-hop Question Answering. In Inui, K.; Jiang, J.; Ng, V.; and Wan, X., eds., *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, 2578–2589. Association for Computational Linguistics.

Yadav, V.; et al. 2020. Unsupervised Alignment-based Iterative Evidence Retrieval for Multi-hop Question Answering. In Jurafsky, D.; Chai, J.; Schluter, N.; and Tetreault, J. R., eds., *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 4514–4525. Association for Computational Linguistics.

Yadav, V.; et al. 2021. If You Want to Go Far Go Together: Unsupervised Joint Candidate Evidence Retrieval for Multi-hop Question Answering. In Toutanova, K.; Rumshisky, A.; Zettlemoyer, L.; Hakkani-Tür, D.; Beltagy, I.; Bethard, S.; Cotterell, R.; Chakraborty, T.; and Zhou, Y., eds., *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, 4571–4581. Association for Computational Linguistics.

Yang, Z.; Qi, P.; Zhang, S.; Bengio, Y.; Cohen, W. W.; Salakhutdinov, R.; and Manning, C. D. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 2369–2380. Association for Computational Linguistics.

You, Y.; Li, J.; Reddi, S. J.; Hseu, J.; Kumar, S.; Bhojanapalli, S.; Song, X.; Demmel, J.; Keutzer, K.; and Hsieh, C. 2020. Large Batch Optimization for Deep Learning: Training BERT in 76 minutes. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.