Towards Graph-hop Retrieval and Reasoning in Complex Question Answering over Textual Database

Minjun Zhu[1,2], **Yixuan Weng** [1], **Shizhu He**[1,2,✉] ,
Kang Liu[1,2,4], **Haifeng Liu**[3], **Yang Jun**[3], **Jun Zhao**[1,2]
[1] The Laboratory of Cognition and Decision Intelligence for Complex Systems, IA, CAS
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
[3] Guangdong OPPO Mobile Telecommunications Corp.,Ltd.
[4] Shanghai Artificial Intelligence Laboratory
minjun.zhu23@gmail.com, {shizhu.he, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

In textual question answering (TQA) systems, complex questions often require retrieving multiple textual fact chains with multiple reasoning steps. While existing benchmarks are limited to single-chain or single-hop retrieval scenarios. In this paper, we propose to conduct **Graph-Hop** —a novel multi-chains and multi-hops retrieval and reasoning paradigm in complex question answering. We construct a new benchmark called **ReasonGraphQA**, which provides explicit and fine-grained evidence graphs for complex question to support comprehensive and detailed reasoning. In order to further study how graph-based evidential reasoning can be performed, we explore what form of Graph-Hop works best for generating textual evidence explanations in knowledge reasoning and question answering. We have thoroughly evaluated existing evidence retrieval and reasoning models on the ReasonGraphQA. Experiments highlight Graph-Hop is a promising direction for answering complex questions, but it still has certain limitations. We have further studied mitigation strategies to meet these challenges and discuss future directions. The code is released at: https://github.com/zhu-minjun/Graphhop.

Keywords: Text Question Answering, Reasoning Graph, Natural Language Database

1. Introduction

Retrieving and reasoning about knowledge is the core ability of question answering (QA) task (Gupta et al., 2019). Textual question answering (TQA) systems retrieve relevant evidence and conduct knowledge reasoning (Chen et al., 2017; Zhu et al., 2022a) when answering complex questions over multiple passages or facts. Considering the flexible form and rich information of text resources, lots of TQA tasks and datasets have been proposed and sparked significant progress in different scenarios (Zhu et al., 2021; Thorne et al., 2021; Li et al., 2024).

However, those datasets still have some limitations. On the one hand, most open domain question answering dataset only focus on multi-hop reasoning of a single chain (Yang et al., 2018; Qi et al., 2021). On the other hand, some textual datasets include multiple discretization chains but only requires single-hop reasoning to answer question, such as WIKINLDB (Thorne et al., 2021) and eQASC (Jhamtani and Clark, 2020).

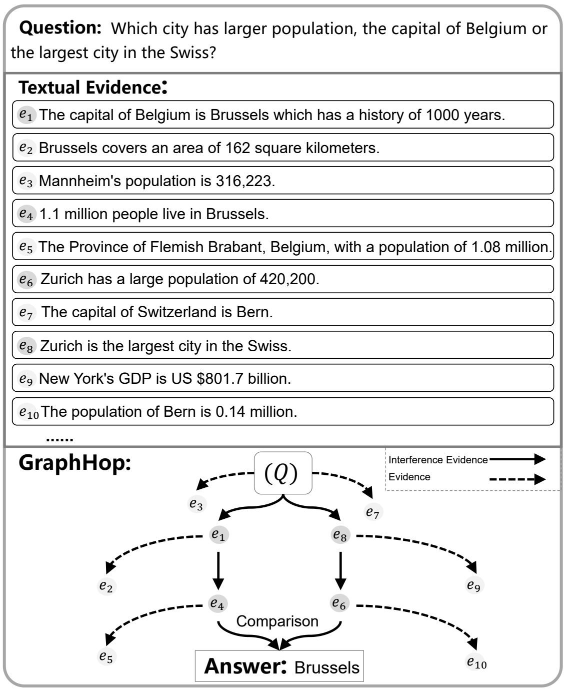
In fact, answering complex questions often requires a combination of retrieving multi-chains and using multi-hops reasoning to infer the answer. As shown in figure 1, to answer this question *"Which city has larger population, the capital of Belgium or the largest city in the Swiss?"*, system should first retrieve the population of each city (multi-chain), and then use multi-hop reasoning on each chain to infer the population value. Finally, it compares two values and identifies the city with the larger population. This process requires an evidence graph with two chains and two hops. We refer to this process as Graph-Hop retrieval and reasoning (shorted as **Graph-Hop**). In this way, Graph-Hop provides a more fine-grained and adaptable representation for complex question answering tasks.

In this work, we introduce a benchmark called ReasonGraphQA and provide interpretable evidence graphs to explicitly describe the reasoning process for solving complex questions. Evidence graphs can provide intermediate reasoning steps and facilitate human understanding. It also allows for better control of the model behavior, enabling users to easily identify errors by inspecting the outputs of intermediate steps. The dataset includes 5 reasoning types and 262 evidence graph structures, while the reasoning path structures are less than 8 in other datasets. We comprehensively evaluate the random samples in terms of text fluency and inference fidelity, and their quality is satisfactory.

Furthermore, we find that retrieving evidence from both forward and backward directions and then fusing them to construct an evidence graph to support answering complex questions is more conducive to generating accurate explanation graphs. We term this approach **B**idirectional **G**raph **R**etrieval (**BGR**). We compared four types of retrieval and reasoning systems on the Rea-

Dataset	Reasoning Types	Evidence	Text Type	Multi-Chains	Multi-Hops	Evidence Structures
TriviaQA (Joshi et al., 2017)	-	✗	Passage	✗	✗	1
HotPotQA (Yang et al., 2018)	3	✓	Passage	✗	✗	1
eQASC (Jhamtani and Clark, 2020)	2	✓	Sentence	✗	✓	1
BeerQA (Qi et al., 2021)	3	✗	Passage	✗	✓	3+
WikiNLDB (Thorne et al., 2021)	4	✓	Sentence	✓	✗	2
ReasonChainQA (Zhu et al., 2022b)	4	✓	Sentence	✓	✗	7
ReasonGraphQA (Ours)	**5**	✓	Sentence	✓	✓	**262**

Table 1: Comparison ReasonGraphQA with existing datasets of Textual Question Answering.



Question: Which city has larger population, the capital of Belgium or the largest city in the Swiss?

Textual Evidence:

e_1 The capital of Belgium is Brussels which has a history of 1000 years.

e_2 Brussels covers an area of 162 square kilometers.

e_3 Mannheim's population is 316,223.

e_4 1.1 million people live in Brussels.

e_5 The Province of Flemish Brabant, Belgium, with a population of 1.08 million.

e_6 Zurich has a large population of 420,200.

e_7 The capital of Switzerland is Bern.

e_8 Zurich is the largest city in the Swiss.

e_9 New York's GDP is US $801.7 billion.

e_{10} The population of Bern is 0.14 million.

......

Figure 1: An example of ReasonGraphQA, it requires multiple chains of fact sets and each chain involves two-hop reasoning in answering this complex question.

sonGraphQA dataset. Experimental results have shown that BGR achieved strong performance in both retrieval and explanation graph tasks. However, their performance is still far from human-level performance in the explanation graph construction task, it is suggested that further research should consider more on Graph-Hop.

In summary, our contributions are as follows: **(1)** We propose a Graph-Hop paradigm and construct a new benchmark ReasonGraphQA, which includes diverse question types and explicit reasoning processes to guide interpretable retrieval and question answering over textual databases in a fine-grained and comprehensive way. **(2)** We also propose a Bidirectional Graph Retrieval (BGR) method, which utilizes both forward reasoning and backward reasoning information, to conduct more efficient and comprehensive reasoning. **(3)** Our evaluation of four retrieval systems on ReasonGraphQA demonstrates that Graph-Hop Retrieval is a promising approach. ReasonGraphQA also shows a challenge for large language model in reasoning ability. We also discuss potential future directions to address Graph-Hop challenges.

2. Related Work

Some researchers proposed a novel QA task over natural language database (NLDB) and support natural language database queries such as filtering, comparison and aggregation, where database is consist of unordered sets of textual facts (Thorne et al., 2021; Zhu et al., 2022b). Each fact is composed of text with different meanings rather than triples that unlike knowledge base QA. It requires comprehensive reasoning and retrieval of text sentences (Wolfson et al., 2020).

Despite the rapid progress in TQA, they ignore the problem of multi-hop retrieval in large-scale facts set. For example in Table 1, eQASC (Jhamtani and Clark, 2020) and BeerQA (Qi et al., 2021) are limited in breadth search, and the WIKINLDB (Thorne et al., 2021) are limited about depth search. In comparison, the proposed ReasonGraphQA requires graph retrieval from large-scale textual databases. And we focus on the discrete reasoning over textual evidences, which greatly evaluate the structured path modeling and discrete reasoning ability of QA systems over textual database. On the other hand, some datasets (Dalvi et al., 2021a) provide a graph-structure-like reasoning process, but only retrieve from a small amount of evidence, limiting the task scenarios.

Existing textual question answering systems still have trouble explaining explicitly why an answer is correct or not and "how" the answer is obtained step-by-step from large-scale facts set (Mou et al., 2021; Rudra et al., 2021; Lu et al., 2020; Trivedi et al., 2022). Although the existing retrieval methods can directly retrieve the relevant passages, they cannot retrieve a structured evidence graph, which limits the ability of the model's reasoning and interpretation (Thorne et al., 2021; Zhu et al., 2022b, 2023).

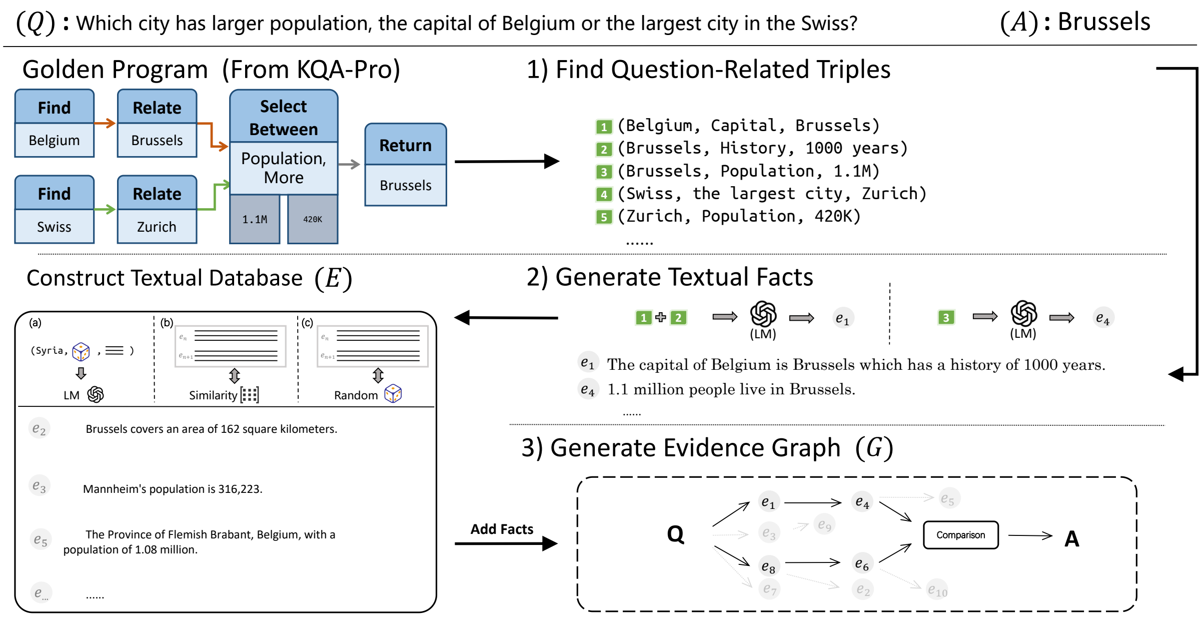

(Q) : Which city has larger population, the capital of Belgium or the largest city in the Swiss?
(A) : Brussels
Golden Program (From KQA-Pro)
Find
Belgium
Relate
Brussels
Select Between
Population, More
1.1M
420K
Find
Swiss
Relate
Zurich
Return
Brussels
1) Find Question-Related Triples
1 (Belgium, Capital, Brussels)
2 (Brussels, History, 1000 years)
3 (Brussels, Population, 1.1M)
4 (Swiss, the largest city, Zurich)
5 (Zurich, Population, 420K)
......
2) Generate Textual Facts
1 + 2
(LM)
e1
3
(LM)
e4
e1 The capital of Belgium is Brussels which has a history of 1000 years.
e4 1.1 million people live in Brussels.
......
Construct Textual Database (E)
(a)
(Syria,
LM
(b)
Similarity
(c)
Random
e2 Brussels covers an area of 162 square kilometers.
e3 Mannheim's population is 316,223.
e5 The Province of Flemish Brabant, Belgium, with a population of 1.08 million.
e... ......
Add Facts
3) Generate Evidence Graph (G)
Q
e1
e4
e5
e3
e9
e8
e6
e7
e2
e10
Comparison
A

Figure 2: ReasonGraphQA construction process. We use Golden Program to generate explanation evidence graphs and create a text database for each question-answer pair. It consisting of three steps: finding question-related triples, generating textual facts, and generating evidence graph.

3. Graph-Hop Over Textual Database

3.1. Data

We present ReasonGraphQA, a new dataset that devotes to answering complex Graph-Hop (multi-hop multi-chain) questions over database. And we also develop an approach to automatically construct a dataset with complex questions, answers and explanation evidence graphs. In this dataset, both question and evidences of database are represented as natural language sentences, each evidence is stand-alone fact. As depicted in Figure 1, formally, given a question Q and a textual database $E = \{e_1, \dots, e_n\}$, system needs to: (1) retrieve an explicable reasoning graph G from the given textual database, (2) obtain the answer A based on the explanation graph G; The graph G is a directed acyclic graph composed of the evidences in E that are related to the question and used to reason the answer.

Figure 2 illustrates the main construction process of ReasonGraphQA using the example in Figure 1. We extend questions and evidence process from single-chain (Zhu et al., 2022b) to much more reason graph structures. We first create 50,000 natural language facts that are needed in answering questions, and construct textual database for each question. Then we aim to automatically acquire evidence graph for graph-hop quastion.

3.1.1. Question-related Triples Finding

Firstly, we obtain complex questions and answers by utilizing a large-scale KBQA dataset (Shi et al., 2022) which requires reasoning over multiple pieces of evidence, and obtain aligned triples. To automate the generation of question-related evidence, we use structured queries “KoPL program” of the KQA-pro dataset and ground each programming procedure to Wikidata triples. As illustrated in Fig. 2 (1), the structured Golden Program, consisting of “Relate”, “Find”, and “Select between” operations, can identify five triples of Wikidata. By searching the target knowledge base (e.g., Wikidata), we can obtain factual facts needed to answer the question.

3.1.2. Textual Facts Generation

Then, we convert structured facts into unstructured texts based on data-to-text work (Agarwal et al., 2021). To improve the diversity, naturalness, and information of the generated text, we propose a method of building triple subgraphs by selecting 0-2 triples with the same head entity from Wikidata according to a certain probability and combine them into a subgraph. While ensuring that they do not overlap with other subgraphs to make sure textual facts remain independent. The subgraphs are then input into a pre-trained language model (T5) fine-tuned on the KELM(Agarwal et al., 2021) corpus to generate unstructured text. As shown in Figure 2 (2). To ensure completeness of entities in the triples, we use string matching to exclude missing text, and use BERTScore (Zhang et al., 2019) to select the most appropriate text evidence from multiple generated options as the correct evidence.

We obtain a large-scale textual database containing generated evidences. For each question, we can retrieve evidence from those large-scale sentences (e.g., more than 100 billion sentences). However, in our experimental environment (500000 sentences in total), we must consider computing efficiency and retrieval cost. Therefore, we have retrieved an appropriate number of sentences from the complete textual database to form a target textual database from which we select evidence for each question. Specifically, apart from the golden evidence, we also retrieve other sentences that are related to the question to form the target textual database. Additionally, to construct a task closer to the real retrieval scene, and to verify knowledge-based reasoning ability, we have added interference evidence to the database. In this paper, the interference-related evidence is obtained from the following three categories of methods (1/3 of each category): (a) SimCSE (Gao et al., 2021) is used to select evidence with similar semantics of the question; (b) We use the same head entity but different relation triples to regenerate evidence sentences; (c) We randomly select other textual evidence.

3.2. Evidence Graph Generation

Last, the reasoning graph of textual evidence is the key component of ReasonGraphQA. We extract and re-summarize the structure among golden triples with the programming language "KoPL program", and utilize network [1] to build the reasoning graph of sentences. In order to ensure the high quality of the evidence graph, we carefully follow these constraints during its construction. (1) Each evidence contains at least one knowledge fact; (2) Each question must be answered with a clear reasoning explanation graph G; (3) Each graph G must be a directed acyclic graph; (4) Any non-leaf node has at least one path to the root node; (5) All evidence cannot be repeated on the path to the root node (avoiding loops). Samples that do not meet these constraints are removed. An example of evidence graph is shown in figure 2. (3), which reflects the reasoning progress from question to answer.

3.3. Dataset Analysis

The ReasonGraphQA dataset consists of 14,678 examples, which are divided into training (11,703), dev (1,506), and test (1,469) sets using a random probability of 8:1:1. Table 2 and Table 3 presents statistics on the graph size and structure of the dataset. The dataset includes four types of evidence graphs: "single-chain single-hop," "single-chain multi-hop," "multi-chain single-hop," and "multi-chain multi-hop," which account for 19.5%, 38.6%, 8.7%, and 33.2% of the dataset, respectively. There are 262 nonisomorphic graph structures in the dataset. As see in Figure **??**, the questions in the dataset are classified into five types: "query", "comparison", "count", "boolean", and "qualifier" based on nine asking strategies used in original KQA-Pro dataset. The "Comparison" involves comparison of multiple evidences. The "Query" type inquires head or tail entity of relational knowledge, the "Qualifier" query for attributes and relations, the "Count" type's answer is number, and the "bool" is to judge correctness of a statement. Most question type involves a variety of graph reasoning. The question types that involve the most graph structures are "Queryname", "Count" and "Queryattribute", which comprehensively involve value comparison, relational knowledge, and time knowledge. This further shows the complexity of our data set.

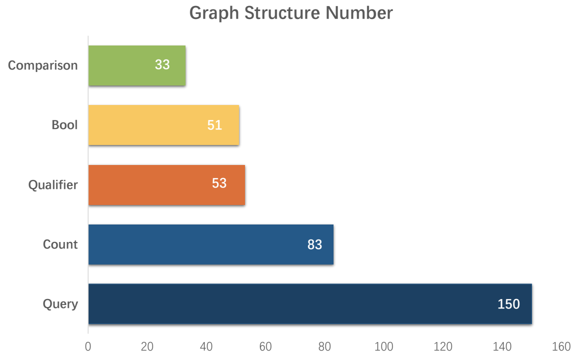


Figure 3: Graph statistics of ReasonGraphQA

3.4. Quality Evaluation

To evaluate the quality of mapping facts from knowledge triples, 2000 sampled facts were scored based on smoothness, faithfulness, and sufficiency. This includes all 1,469 test set samples and an additional 531 training set samples. 98.3% (1966/2000) facts were smooth, with only 34 containing repeated text. 98.9% (1978/2000) facts were faithful to the relation of the triples, with only 22 containing additional information. We found that the quality of the data set construction is relatively high. For example, 96% of facts in WiKiNLDB (Thorne et al., 2021) are loyal to relationships, while ReasonGraphQA is 98.9%. This demonstrates that the data set presented in this paper is suitable for model development and technical verification of complex question answering in textual

[1] https://networkx.org

Dataset	Average Chain	Average Hop	Average Candidate Facts	Max Chain	Max Hop	Max Facts
ReasonGraphQA	1.56	2.10	3.28	5	23	24

Table 2: The statistics of reasoning explanation graphs in the ReasonGraphQA dataset

Dataset	SC,SH	SC,MH	MC,SH	MC,MH	Number
Train	2,295	3,524	1,001	3,883	11,703
Dev	321	577	141	467	1,506
Test	248	572	135	514	1,469
Total	2,864	5,673	1,277	4,864	14,678

Table 3: The statistics of ReasonGraphQA, where SC, MC, SH and MH indicate single-chain, multi-chain, single-hop, and multi-hop, respectively.

databases. After the evaluation was completed, we manually corrected the samples that contained errors, thereby ensuring the accuracy of the evaluation data (test set samples).

4. Methods

In this section, we present our proposed retrieval-based question-answering model. This model follows the popular retrieval-reader architecture. Figure 4 illustrates the architecture of our model, which consists of Bidirectional Graph-Hop Retrieval, Subgraph Reconstruction, and Answer Generation.

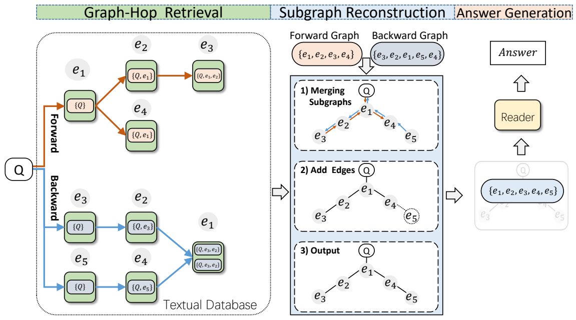


Figure 4: Overview of our proposed Bidirectional Graph-hop Retrieval (BGR) method.

4.1. Bidirectional Retrieval

We design a bidirectional retrieval method to improve graph-hop retrieval accuracy. In traditional chain retrieval, the model starts by searching for the first relevant evidence, and continues iteratively. However, the structure of evidence in graph retrieval is more complex, resulting in a higher error rate as the search depth increases. To mitigate this issue, we introduce backward retrieval. ReasonGraphQA provides reason graph structures, so we define the reason process from question to answer as forward, and the process from answer to question as backward. In the BGR model, we use two separate BERT models to model the forward and backward processes respectively, and their weights are not shared. As a result, we obtain two evidence subgraphs,one from forward retrieval and one from backward retrieval, as depicted in Figure 4. By merging these two subgraphs, our bidirectional retrieval method can mitigate the problem of rapidly declining accuracy in the forward retrieval with increasing depth in the graph.

Given a question Q and a candidate evidence base $E = \{e_1, e_2, \cdots, e_n\}$, we represent Q and E using BERT to obtain their representations, $h_0 = \text{BERT}(Q)$. The retrieval process follows a depth-first search, where at each step, the current evidence node i may have multiple paths that are reachable. These paths are represented as $H_i = \{h_i^1, \cdots, h_i^{i_k}\}$, where i_k is the number of paths per node. These paths are matched one by one with the path code and evidence base E_i ($E_i \subseteq E$). To handle the complex structure of graph retrieval, we use a feedforward neural network (composed of linear layers and activation functions) instead of a similarity threshold to match the next layer of evidence nodes. Every time a new evidence node is retrieved, we use the Attention mechanism to combine the path set H_i and the retrieved evidence node e_{i+1} to generate a new path set H_{i+1}. The whole process is illustrated in Figure 4. We repeat this process until no new nodes can be retrieved.

4.2. Subgraph Reconstruction

The reconstruction process of the evidence graph is depicted in Figure 4. We utilize networkx[2] to build two subgraphs using forward and reverse retrieval techniques. Reverse retrieval allows us to verify the accuracy of our findings. By intersecting the edges of the two subgraphs and removing any non-overlapping nodes and edges, we can construct a complete evidence graph. This evidence graph visually demonstrates the reasoning process from the initial question to the final answer.

We first propose a bidirectional subgraph confidence score (BSC), which can be used to evaluate the degree of confidence between two subgraphs. We extract edges from the forward subgraph and the backward subgraph respectively, and then select by evaluating the BSC of the subgraph. If the BSC is less than γ, intersection of the bidirectional

[2] https://networkx.org/

subgraphs is taken to reconstruct the graph, and new edges are not added twice for the existing nodes.

$$BSC(G_F, G_B) = \frac{Edge_F \cap Edge_B}{Edge_F \cup Edge_B} \quad (1)$$

$$G = \begin{cases} G_F \cup G_B & \text{if BSC}(G_F, G_B) > \gamma \\ G_B & \text{if BSC}(G_F, G_B) \leq \gamma \end{cases} \quad (2)$$

where $Edge_F$,$Edge_B$ is the edge set of forward and backward subgraghs. If the BSC is greater than γ, the backward subgraph is reserved. A threshold value of γ is used to determine whether the intersection of the two subgraphs should be used to construct the final evidence graph. The reason for using BSC and threshold value γ is that, it can effectively improve the retrieval performance, by preserving the integrity and accuracy of the final evidence graph, also it can help to prevent from adding unnecessary edges.

4.3. Answer Generation

In order to generate an answer, the multiple evidences are fed into the reader as following.

$$A = Reader_{T5}\left(\{e_i | e_i \in G\}\right) \quad (3)$$

where evidences are ordered according to the structure of the retrieved evidence graph G.

To measure the retrieval performance, We follow the previous settings (Thorne et al., 2021; Zhu et al., 2022b) and use the classic T5 (Raffel et al., 2019) model as the fixed reader, but this can easily be adapted to other pre-trained language models.

5. Experiments

In this section, we analyze the performance of different retrieval and reasoning systems on ReasonGraphQA, and investigate performance and limitations of our proposed graph-hop retrieval system.

5.1. Compared Baselines

We compare retrieval models of two retrieval mechanisms representative. Single-Hop retrieval method that retrieves all evidence at once (Random, BM25 (Amati, 2009), DPR (Karpukhin et al., 2020)). Multi-hop retrieval methods retrieve one evidence iteratively in one step (GRR (Asai et al., 2020), MDR (Xiong et al., 2021), SSG (Thorne et al., 2021)). We use the code and parameter settings provided by the original papers for all baselines. For single-Hop retrieval models (BM25, DPR, SSG), we retrieve the top-k evidence, where k is the size of the golden evidence set.

We also explore the potential of large language models (LLM) in solving complex reasoning tasks through few-shot learning (Wei et al., 2022; Weng et al., 2022, 2023). To this end, we have developed five reasoning graph prompts for LLM. These prompts aim to enable the construction of a reason graph by LLM.

All methods are tested in the Dev set at the end of each round, and the model with the highest retrieval accuracy in the Dev set is selected for testing. We repeate the process three times by replacing the random seeds and average them as the final result.

5.2. Implementation

To measure retrieval mechanism in a fairer open-domain setting, We uniformly use T5-base model (Raffel et al., 2019) as reader, and input retrieval evidence of different methods into a fine-tuned T5 model to generate answer. A bert-base-uncased model is chosen as text encoder for extracting feature. We use AdamW (Loshchilov and Hutter, 2018) with warm-up as the optimizer. The learning rate, epoch and batch size are set to 1×10^{-5}, 20, 8 respectively. Text maximum length n was set as 30 and the d was set as 768.

5.3. Evaluation Metrics

In retrieval task, correctness were measured in terms of Explanation Graph and Evidence Set. Following previous works (Yang et al., 2018; Dalvi et al., 2021b), Exact Match (EM) , Precision, Recall and F1 was adopted. As for Explanation Graph evaluation, we used three indicators, Graph Matching (GM) evaluates whether the retrieved evidence graph is consistent with golden evidence graph. Graph Structure (GS) evaluates whether retrieved graph structure and golden graph structure are isomorphic, it will ignore nodes accuracy. Graph Editing Distance (GED) (Abu-Aisheh et al., 2015) measures how many steps does converting retrieved evidence graph to the golden one need. Then we use EM to measure the performance of QA task.

5.4. Results and Analysis

The graph structure and the set retrieval both play a critical role. As shown in Table 4, single-hop methods like DPR perform well in set recall and QA, while multi-hop methods like SSG excel in graph accuracy and QA. This highlights the importance of both the evidence graph structure and set retrieval for accurate question answering. This suggests that previous datasets (Qi et al., 2021), which only evaluate the accuracy of the retrieved set, are not sufficient for measuring QA performance. Additionally, as Table 5 shows, incorpo-

Method		Explanation Graph			Evidence Set				QA EM
		GM↑	GS↑	GED↓	F1↑	Precision↑	Recall ↑	EM↑	Acc↑
Single-Hop Retrieval	Random	-	-	-	-	-	13.29	-	37.51
	BM25 (Amati, 2009)	-	-	-	-	-	70.84	-	62.42
	DPR (Karpukhin et al., 2020)	-	-	-	-	-	88.04	-	67.39
Multi-Hop Retrieval	GRR (Asai et al., 2020)	24.92	25.19	5.86	71.45	**99.39**	60.13	25.05	55.20
	SSG (Thorne et al., 2021)	34.72	35.12	6.43	75.81	78.23	77.94	53.85	63.04
	MDR (Xiong et al., 2021)	25.46	25.46	5.82	84.72	97.96	79.97	62.83	51.26
LLMs' Retrieval	GPT-3 (Brown et al., 2020)	0.07	17.14	8.04	12.75	23.97	10.62	0.07	37.12
	GLM (Zeng et al., 2022)	0.68	4.76	7.09	11.16	21.26	8.67	0.68	38.02
	Instruct-GPT (Ouyang et al., 2022)	35.91	54.77	**1.18**	71.92	67.47	81.79	40.78	56.49
Graph-Hop Retrieval	*only w/* Graph-Hop's Forward	27.71	28.86	6.72	88.78	87.2	92.81	67.12	69.71
	only w/ Graph-Hop's Backward	56.57	57.86	4.64	85.67	85.12	88.55	64.06	67.60
	BGR	**56.71**	**58.48**	4.70	**91.81**	90.785	**95.23**	**68.82**	**70.18**
Human Bound		92.15	93.15	0.18	98.13	98.73	97.54	96.41	95.13

Table 4: Experimental results of BGR compared with three types of Retrieval methods on Retrieval-Reader architecture. We report the results of human in the test set to show the upper bound of human.

Model	W/O Reason Graph	With Reason Graph
GPT-3	1.05	**23.55**
Instruct-GPT	12.43	**45.15**
GLM-130B	4.46	**7.15**
Llama-2-70B	4.12	**17.10**

Table 5: The Zero-shot performance of large language model (GPT-3: `code-davinci-001` Instruct-GPT: `code-davinci-002`) in ReasonGraphQA. We use a method similar to Chain of Thought to add the diagram structure to the input of LLM.

rating graph structure information into evidence results can significantly improve QA performance when using large language models.

LLM is capable of constructing inference diagrams. Large language models have demonstrated performance far surpassing previous models on many question-answering tasks (Zhao et al., 2023; Weng et al., 2024). We evaluated the performance of the original GPT-3 (Brown et al., 2020), the Instruct-GPT (Ouyang et al., 2022), GLM-130B (Zeng et al., 2022) and Llama-2-70B (Touvron et al., 2023) on the ReasonGraphQA datasets. We conducted all experiments in the few-shot setting, without any fine-tuning of the original language model. Apart from the context, we have not provided any other prompt text. We utilize the phrase "Then" to denote the relationship between adjacent nodes, and "On the other hand" to indicate the relationship between different chains. In our LLM retrieval, as shown in Table 4, we discovered that while LLM has a low accuracy rate for the evidence set, it surpasses existing multi-hop retrieval in constructing inference graphs (especially for Instruct-GPT, Graph reasoning ability is close to Graph-Hop) which illustrates the reasoning potential of LLMs, which may be an important direction of future Graph-Hop research.

Graph-Hop is more appropriate for ReasonGraphQA. We note that multi-hop retrieval systems have high precision but low recall, as true nodes at the same level are ignored when retrieving along one reasoning chain. However, BGR can improve recall to 95.227% by utilizing a bidirectional retrieval architecture. Additionally, Graph-Hop's Forward is better in evidence retrieval, while Backward has a higher graph construction capability. In the next section, we will further analyze Graph-Hop's performance and explain why BGR's performance is better after subgraph reconstruction.

5.5. Ablation Study

Bidirectional Retrieval. To better understand cooperation mechanism of Forward retrieval and the Backward retrieval. We perform ablation study on retrieval direction. In Table 4 we can clearly find that backward retrieval has a higher performance in the explanation graph, and forward retrieval has a higher performance in evidence retrieval. The BGR has better performance in explanation graph task, evidence retrieval task. And BGR outperform both forward and backward in QA task. This shows that bidirectional subgraph reconstruction(BCD algorithm) can make up deficiency of both and achieve a balance.

Bidirectional BGR with balanced γ value performs best. As depicted in figure 5, we analyzed the effect of the value γ on the accuracy of retrieving the evidence set and graph. When γ=1, the final evidence graph is G_B. When $\gamma = 0$, the evi-

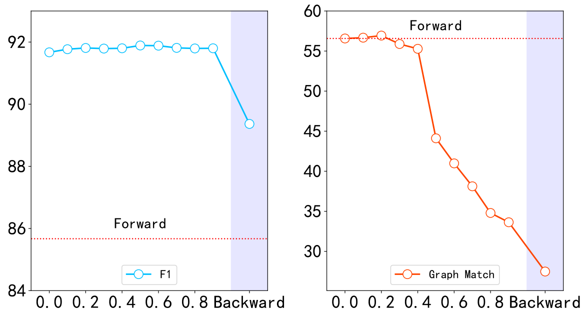


Figure 5: F1 and GM changes with different γ.

dence graph of samples that BSC $\neq 0$ is $G_F \cup G_B$. We found that the accuracy of bidirectional BGR is higher than that of forward and backward BGR, because G_F performs better in graph structure, while G_B tends to retrieve more accurate evidence sets, and the introduction of γ achieves a balanced result in the evidence set and graph structure.

While BGR has achieve strong performance, its still an on-going challenge for graph-hop QA task. This is a meaningful task that are expected to promote development of TQA in knowledge reasoning and interpretability.

5.6. Futher Analysis on ReasonGraphQA.

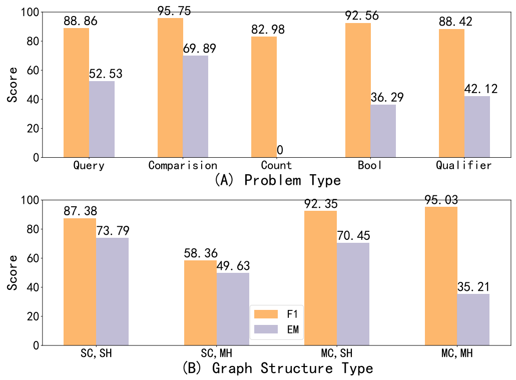


Figure 6: The retrieval performance for different question and graph structure types.

BGR adapts to different question types. We divide the test set into 5 different question types. Figure 6(A) shows detailed accuracy of We can find that the evidence retrieval ability of the BGR can adapt to different kinds of questions, especially "Comparison" and "Bool". However, when faced with the task of constructing evidence graph, it is easy to miss nodes and edges. Even in the "Count" question, the BGR cannot correctly predict any explanation graph. We believe that one of the main reasons is the flaws in language models when it comes to retrieval and QA regarding numerical data. This is a research direction that deserves improvement in the future. This proves that the graph construction task still has a certain complexity, and the BGR still has a large room for improvement in the construction of retrieval evidence graphs.

BGR performs well in complex, multi-hop explanation graph structures. We classify and compare according to the graph structure, which are single-chain single-hop, single-chain multi-hop, multi-chain single-hop, and multi-chain multi-hop. In Figure 6, more complex structure graph show the better retrieval performance, which proves that BGR can efficiently retrieve evidence in complex text question answering. In addition, BGR has achieved the best performance in MCMH explanation graph structures compared with the other three types, which even close to the QA accuracy with perfect retrieval. It shows that BGR is suitable for graph-hop retrieval. However, the more complex the graph structure is, the more edges there are. We believe that the modeling between edges is challenging due to the high similarity of edges between different nodes, which encourages researchers to conduct further research on explanation graph retrieval in the future.

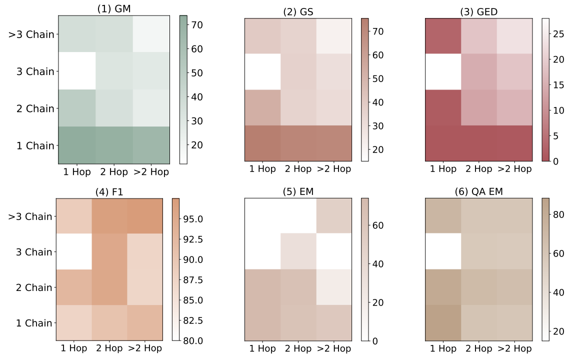


Table 6: Experimental results of BGR at different hops and different chain numbers.

The construction of multi-chain and multi-hop explanation graph is still challenging. We have evaluated how varying hop and chain number of evidence graph structure influencing graph structure (GM, GS, GED), evidence set (F1, EM), and question answering (QA EM). Our findings reveal that retrieving evidence graphs and answering questions from more complex evidence structures remains a challenging task. Specifically, as shown in Figure 6, the graph structure performance of evidence graph retrieval is strong for simple graphs but poor for complex ones, and the Exact Match of evidence sets retrieval is poor in complex graph structures. This results in relatively lower performance in question answering for complex graph structure samples.

6. Conclusion

Our study introduces the ReasonGraphQA dataset, the first textual database QA dataset with an explanation graph, which provides complex structured retrieval assistance for graph retrieval systems. We have tested various traditional evidence retrieval methods on the ReasonGraphQA dataset and evaluated them manually. Additionally, we propose the graph-hop retrieval paradigm and develop a bidirectional graph retrieval model, which significantly improves the evidence retrieval and graph construction capabilities of complex question answering by reconstructing reasoning paths in different directions. Future research utilizing the ReasonGraphQA dataset can enable fine-grained analysis of the explanation graph output from models, leading to further advancements in real and complex QA environments. While the current methods have several limitations, This presents opportunities for future research to improve upon them.

7. Acknowledgements

This work was supported by the National Key R&D Program of China (No.2022ZD0118501) and the National Natural Science Foundation of China (No.62376270, No.62171183), Youth Innovation Promotion Association CAS, and OPPO Research Fund. We also appreciate Bin Li's guidance on paper polishment.

8. Bibliographical References

Zeina Abu-Aisheh, Romain Raveaux, Jean-Yves Ramel, and Patrick Martineau. 2015. An exact graph edit distance algorithm for solving pattern recognition problems. *international conference on pattern recognition applications and methods*.

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3554–3565, Online. Association for Computational Linguistics.

Giambattista Amati. 2009. *BM25*, pages 257–260. Springer US, Boston, MA.

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*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

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.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021a. Explaining answers with entailment trees. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7358–7370, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. 2021b. Explaining answers with entailment trees. *empirical methods in natural language processing*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nitish Gupta, Kevin Lin, Dan Roth, Sameer Singh, and Matt Gardner. 2019. Neural module networks for reasoning over text. *Learning*.

Harsh Jhamtani and Peter Clark. 2020. Learning to explain: Datasets and models for identifying valid reasoning chains in multihop question-answering.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering.

Bin Li, Bin Sun, Shutao Li, Encheng Chen, Hongru Liu, Yixuan Weng, Yongping Bai, and Meiling Hu. 2024. Distinct but correct: generating diversified and entity-revised medical response. *Science China Information Sciences*, 67(3):1–20.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jing Lu, Gustavo Hernandez Abrego, Ji Ma, Jianmo Ni, and Yinfei Yang. 2020. Neural passage retrieval with improved negative contrast.

Xiangyang Mou, Mo Yu, Shiyu Chang, Yufei Feng, Li Zhang, and Hui Su. 2021. Complementary evidence identification in open-domain question answering. *conference of the european chapter of the association for computational linguistics*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Peng Qi, Haejun Lee, Tg Sido, and Christopher Manning. 2021. Answering open-domain questions of varying reasoning steps from text. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3599–3614, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Koustav Rudra, Zeon Trevor Fernando, and Avishek Anand. 2021. An in-depth analysis of passage-level label transfer for contextual document ranking. *arXiv: Information Retrieval*.

Jiaxin Shi, Shulin Cao, Liangming Pan, Yutong Xiang, Lei Hou, Juanzi Li, Hanwang Zhang, and Bin He. 2022. Kqa pro: A dataset with explicit compositional programs for complex question answering over knowledge base.

James Thorne, Majid Yazdani, Marzieh Saeidi, Fabrizio Silvestri, Sebastian Riedel, and Alon Halevy. 2021. Database reasoning over text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3091–3104.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Yixuan Weng, Shizhu He, Kang Liu, Shengping Liu, and Jun Zhao. 2024. Controllm: Crafting diverse personalities for language models.

Yixuan Weng, Zhiqi Wang, Huanxuan Liao, Shizhu He, Shengping Liu, Kang Liu, and Jun Zhao. 2023. Lmtuner: An user-friendly and highly-integrable training framework for fine-tuning large language models. *arXiv preprint arXiv:2308.10252*.

Yixuan Weng, Minjun Zhu, Shizhu He, Kang Liu, and Jun Zhao. 2022. Large language models are reasoners with self-verification. *arXiv preprint arXiv:2212.09561*.

Tomer Wolfson, Mor Geva, Ankit Gupta, Matt Gardner, Yoav Goldberg, Daniel Deutch, and Jonathan Berant. 2020. Break it down: A question understanding benchmark. *Transactions of the Association for Computational Linguistics*.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *empirical methods in natural language processing*.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *Learning*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

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: Artificial Intelligence*.

Minjun Zhu, Bin Li, Yixuan Weng, and Fei Xia. 2022a. A knowledge storage and semantic space alignment method for multi-documents dialogue generation. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 130–135, Dublin, Ireland. Association for Computational Linguistics.

Minjun Zhu, Yixuan Weng, Shizhu He, Kang Liu, and Jun Zhao. 2022b. Reasonchainqa: Text-based complex question answering with explainable evidence chains. *arXiv preprint arXiv:2210.08763*.

Minjun Zhu, Yixuan Weng, Shizhu He, Cunguang Wang, Kang Liu, Li Cai, and Jun Zhao. 2023. Learning to build reasoning chains by reliable path retrieval. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.