Decomposing Complex Questions Makes Multi-Hop QA Easier and More Interpretable

Ruiliu Fu[1,2] and **Han Wang**[1,2] and **Xuejun Zhang**[1,2] and **Jun Zhou**[1] and **Yonghong Yan**[1]
[1]Institute of Acoustics, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
{furuiliu,wanghan,zhangxuejun,zhoujun,yanyonghong}@hccl.ioa.ac.cn

Abstract

Multi-hop QA requires the machine to answer complex questions through finding multiple clues and reasoning, and provide explanatory evidence to demonstrate the machine's reasoning process. We propose **R**elation **E**xtractor-**R**eader and **C**omparator (RERC), a three-stage framework based on complex question decomposition. The Relation Extractor decomposes the complex question, and then the Reader answers the sub-questions in turn, and finally the Comparator performs numerical comparison and summarizes all to get the final answer, where the entire process itself constitutes a complete reasoning evidence path. In the 2WikiMultiHopQA dataset, our RERC model has achieved the state-of-the-art performance, with a winning joint F1 score of 53.58 on the leaderboard. All indicators of our RERC are close to human performance, with only 1.95 behind the human level in F1 score of support fact. At the same time, the evidence path provided by our RERC framework has excellent readability and faithfulness.

1 Introduction

Multi-hop QA is an important and challenging task in natural language processing (NLP), which requires complex reasoning over several paragraphs to reach the final answer and explanatory evidence to demonstrate the reasoning process. Many high-quality multi-hop QA datasets have been introduced recently, such as HotpotQA (Yang et al., 2018), ComplexWebQuestions (Talmor and Berant, 2018), QAngaroo WikiHop (Welbl et al., 2018), R^4C (Inoue et al., 2020), 2WikiMultiHopQA (Ho et al., 2021), etc.

These high-quality multi-hop QA datasets promote many multi-hop QA models (Song et al., 2018; Ding et al., 2019; Xiao et al., 2019; Nishida et al., 2019; Tu et al., 2019; Cao et al., 2019), most of which are end-to-end models based on graph structure or graph neural network (Veličković et al., 2018). Although these works have good performances in many tasks, they also have some limitations to address. First of all, the internal reasoning mechanism of previous end-to-end QA models is a black-box, which usually use an additional discriminator to judge whether a sentence is a clue sentence, such as DFGN (Xiao et al., 2019). There is no evidence to show that such additional discriminators are strongly correlated with the reasoning results of the end-to-end model, which means not faithful. Secondly, although graph structure is helpful to multi-hop reasoning in theory, but recent work (Shao et al., 2020) shows that the existing graph neural network is only a special attention mechanism (Bahdanau et al., 2014), and it's not necessary for multi-hop QA, with the experiments that better results can be achieved by using only transformer network instead of graph neural network, as long as the same additional adjacency matrix information is provided.

We observed that human reasoning about complex questions is not accomplished overnight and it's usually divided into the steps of question decomposition, answering sub-questions, summarizing and comparing. For example, for the complex question, "whose candidate will get more votes in the 2020 U.S. election, Democrats and Republicans?" People will not think about the whole question, but firstly decompose the complex question. Realizing that the subject of the question is "Democrats and Republicans", and the question is about "candidates" and "number of votes", people can answer those sub-questions progressively – "who is the Democratic candidate?" and "how many votes does **ANS** get?" The same thinking process was performed for another question subject, "Republican Party". Finally, the two votes were compared to obtain the answer to the entire complex question.

Inspired by the way humans answer complex multi-hop questions, in this work we abandoned

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 169–180
November 7–11, 2021.

the end-to-end model structure, but imitated the human reasoning mechanism to propose a three-stage Relation Extractor-Reader and Comparator (RERC) model[1]. We first build a Relation Extractor, which can automatically extract the subject and key relations of the question from the complex unstructured textual representation. For the Relation Extractor, we use two different structures, one is classification-type (CRERC), where the evidence relation information in the dataset is used as prior knowledge, and the question text is mapped to question relations through the classifier; the other is span-type (SRERC), where the type of question relations is unrestricted, and the Relation Extractor can automatically extract multiple corresponding spans from the question text as question relations. Next, we use the advanced ALBERT model (Lan et al., 2020) as the Reader, which reads the corresponding paragraphs and answer each sub-question composed of the subject and relations of the question in turn. Finally, for comparison type questions, our Comparator module compares the magnitude of each subject's final answer, and then get the entire answer.

Our contributions are summarized as follows:

- We propose a novel RERC model for multi-hop text-based QA and evidence path search tasks.
- We propose a Query-aware Entity Tree Paragraph Screening (QETPS) method to filter valid paragraphs from a large number of documents before Reader module, which is more efficiently than previous paragraph selecting methods.
- We provide an experimental study on a public multi-hop dataset (2WikiMultiHopQA) to demonstrate that our proposed RERC model has the state-of-the-art performance in both answering multi-hop questions and extracting evidence at the same time.

2 Related work

2.1 Multi-hop QA research

Initially, researchers still has been using the previous ideas in single-hop reading comprehension, focusing on the query-document co-inference attention method (Dhingra et al., 2018; Zhong et al., 2019; Cao et al., 2019). Until Ding et al. (2019) cleverly applied the graph neural network to the multi-hop QA task, and achieved excellent performance improvement, then other models such as DFGN (Xiao et al., 2019) were successively proposed to integrate graph structure into multi-hop QA tasks.

However, recently these end-to-end methods in multi-hop QA tasks seem to have fallen into a bottleneck that there is still a huge gap from human level. Besides, the internal reasoning process of these end-to-end multi-hop QA models is not clear, and the generated explanations are not faithful enough. Our proposed Relation Extractor-Reader and Comparator (RERC) model adopts the idea of decomposing complex questions. It decomposes complex multi-hop QA tasks into multiple single-hop reading comprehension subtasks, and transforms complex tasks into simple tasks that we have solved. In this way, the RERC model has successfully avoided the dilemmas of unclear internal mechanism and unfaithful interpretation caused by the separation of interpretation and reasoning, which the above-mentioned existing end-to-end models have faced.

2.2 Complex question decomposition

Complex question decomposition is also an important task in NLP area, which is closely related to multi-hop QA task. For example, the DecompRC model (Min et al., 2019) regarded the complex question decomposition as a span extraction task, and used a supervised model to decompose the complex question into multiple spans to solve the multi-hop QA task. However, this method of using question spans as sub-questions is only suitable for specific Compositional-type complex questions. Not all complex questions can be decomposed into sub-questions by question fragments. ONUS (Perez et al., 2020) adopted an unsupervised method, using the characteristics of HotpotQA (Yang et al., 2018) multi-hop QA dataset and SQuAD (Rajpurkar et al., 2016, 2018) single-hop reading comprehension dataset which are both based on Wikipedia document, and used similar matching to construct some pseudo-data from complex questions to simple questions, and then trained an unsupervised sequence-to-sequence (seq2seq) model (Artetxe et al., 2018) to generate sub-questions. The method relies on the homology characteristics of the two datasets HotpotQA (Yang et al., 2018) and SQuAD (Rajpurkar et al., 2016,

[1]Our source code is available in https://github.com/furuiliu/RERC.

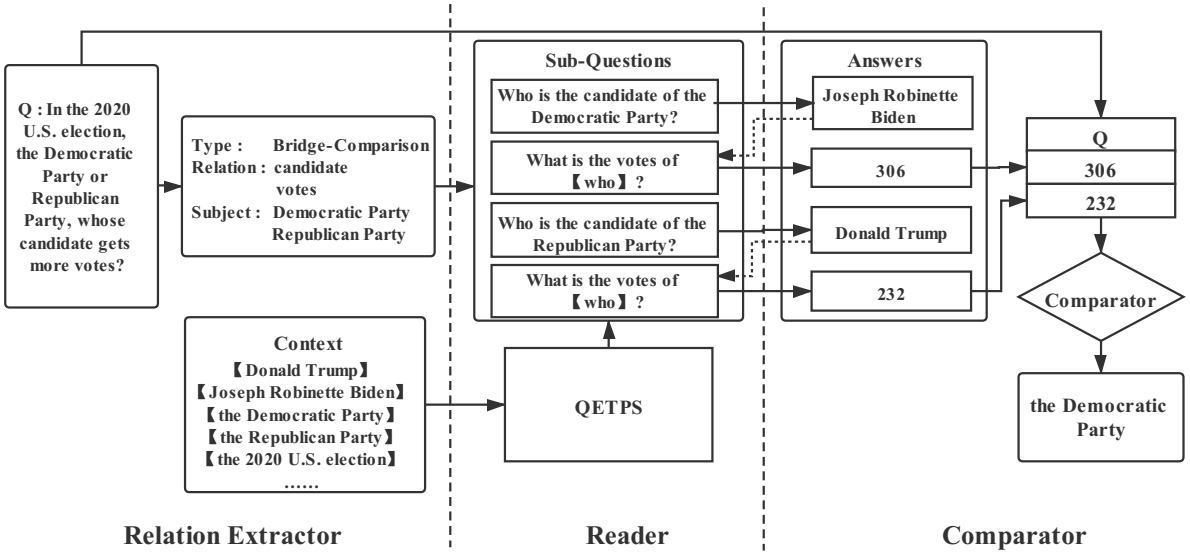

Figure 1: Relation Extractor-Reader and Comparator (RERC) model

2018), which is more restrictive.

In this work, we propose that the complex question decomposition model used for multi-hop QA does not need to generate complete sentence-type sub-questions and most complex questions cannot be directly divided into complete sub-questions. We only need to extract the key question subjects and question relations, and then construct them through templates, which not only reduce the difficulty of decomposition of complex questions, but also apply to the decomposition of complex questions of any form.

3 Proposed method

In this section, we will introduce in detail our proposed Relation Extractor-Reader and Comparator (RERC) model. This is a three-stage multi-hop QA model consisting of three parts: Relation Extractor, Reader and Comparator. The working principle of the whole framework is shown in figure 1. Given the question q and the context set $C = \{c_i\}$, firstly pass the question q to the Relation Extractor to obtain the question subjects set $E = \{e_i\}$ and the question relations set $R = \{r_i\}$, and then construct the sub-questions set $SQ = \{sq_i\}$, and then the Reader reads the searched context and answers each sub-question to obtain the answer set $A = \{a_i\}$, and finaly the Comparator obtains the final answer ANS through numerical comparison and summary analysis.

3.1 Relation Extractor

In this work, we have experimented with two Relation Extractors, named Classification-type Relation Extractor (CRE) and Span-type Relation Extractor (SRE). The difference between the two Relation Extractors is whether to use the evidence relation information in the dataset, so they are only distinguished in the output layer.

The Classification-type Relation Extractor (CRE) is firstly introduced, which uses an advanced text classifier structure. We first use the advanced large-scale pre-training language model BERT (Devlin et al., 2018) to encode the question q to obtain a question encoding representation $Q_{Embed} \in R^{l \times d}$ with rich semantic information.

Next, we need to perform self-interaction calculations on the entire sentence and find the relationship between the words in the sentence through the self-attention mechanism, so as to find the key information corresponding to the subject and the relations of the question. We use the Transformer network based on the self-attention mechanism (Vaswani et al., 2017) as our interaction layer, by encoding the question representation Q_{Embed} obtained above to express self-interaction, and then get the question self-interaction representation $Q_{Inter} \in R^{l \times d}$ and the question pooling representation $Q_{Pooled} \in R^{1 \times d}$ after MaxMeanPooler pooling operation.

$$Q_{Inter} = Transformer(Q_{Embed}) \quad (1)$$

In order to make specific reasoning for different

question types (Compositional, Inference, Comparison and Bridge-Comparison), we need to determine the question type $Type$ first. Therefore, we deploy a linear classification layer $TypeLinear$ to calculate the probability of the four question types $Q_{Type} \in R^{1\times4}$ and category prediction $T = argmax(Q_{Type})$:

$$Q_{Type} = TypeLinear(Q_{Pooled}) \quad (2)$$

For the four different question types, we use four independent relation classifiers $\{RelationLinear_i\}$, and then use the category-aware fusion mechanism to fuse the results of the four relation classifiers to get the final relation prediction result R:

$$R = argmax(Q_{Type} \cdot r) \quad (3)$$

where $r = [r_1, r_2, r_3, r_4] \in R^{4\times2\times n}$, $r_i = RelationLinear_i(Q_{Pooled})$, $i = 1, 2, 3, 4$.

Besides, we also predict the question subject entity, which is a sequence span extraction task. We choose a pointer network (Vinyals et al., 2015) EntityPointer to perform this task:

$$E = EntityPointer(Q_{Inter}) \quad (4)$$

The loss function of the Relation Extractor is designed as $loss = loss_R + \alpha \cdot loss_T + \beta \cdot loss_E$, where $loss_R$, $loss_T$ and $loss_E$ represent the prediction loss of the question relation, question type, and the question subject respectively.

Above is the detailed structure of the entire Classification-type Relation Extractor (CRE). However, the CRE model must be required to limit the known relation categories, which greatly limits its versatility. Therefore, we additionally propose a Span-type Relation Extractor (SRE) to replace relation category prediction with relation span extraction. We also use four pointer networks $\{RelationPointer_i\}$ to perform relation span extraction, and then perform category-aware fusion. The whole process is basically the same as the CRE model, so we don't repeat it here.

After obtaining the prediction results of the subjects and the relations of the question, they are spliced together to form the sub-questions set $SQ = \{sq_i\}$ which are sent to the next Reader module.

$$SQ = \{e_i | r_j \ \forall e_i \in E, r_j \in R\} \quad (5)$$

3.2 Sub-Question Reader

Before reading comprehension, we need to sort or filter all the paragraphs, because our model has a limited ability to process long-sequence texts, and the total length of the context in the task greatly exceeds this limit, which is also common in practical applications, and most of context is useless to answer the sub-questions. We propose a Query-aware Entity Tree Paragraph Screening (QETPS) method.

Through careful observation, we find that every hop in the multi-hop QA dataset needs to pass through the entity (person, organization, location, etc.) as a transfer, which is also in line with our common sense of life. Therefore, we can build an entity tree through the interdependence between entities to make each paragraph sorted according to priority.

Specifically, we first locate all entities in the question sentence and use these entities as the root nodes of the entity tree. Then we look for the paragraphs where these root entities appear, and associate those entities that appear in the same sentence with root entities as the child nodes. Then we start from these child nodes and repeat the above process until no new child nodes can be added to the tree, at this time our entity tree is formed. In order to prevent the influence of interfering paragraphs, we have added a query-aware regulation mechanism that only the child nodes in the corresponding sentence of the query can be added. At the same time, in order to ensure the effectiveness of the method, we did not use exact matching(EM) when searching for the corresponding entities or relations. Instead, we used the F1 value calculated by the longest common subsequence length as the similarity, by setting threshold to determine whether it appears.

After constructing the entity tree, we believe that the answer for the ith-hop sub-question is most likely to exist in the paragraph associated with the node at the ith level of the entity tree (the root node is the 0th level). So we successively obtain the filtered paragraph representation C_{QETPS} through adding paragraphs corresponding to nodes according to the distance in the tree.

Next, we use the advanced AlbertForQuestionAnswering model (Lan et al., 2020) as Reader to answer each sub-question, and get the answer set $A = \{a_i\}$:

$$a_i = Reader(sq_i \mid C^i_{QETPS}), \ i = 1, 2, 3, ... \quad (6)$$

3.3 Comparator

After getting the answers to all sub-questions, we need to summarize these answers to get the final answer, which also depends on the question type we get in the Relation Extractor. For Compositional-type and Inference-type questions, we only need to output the answer of the last sub-question. So we should focus on Comparison-type and Bridge-Comparison-type questions.

We trained a Comparator that can compare various types of quantitative relationship problems universally. We splice the question text description and the two objects to be compared, and send them to the quantity relationship Comparator to get the comparison result $A_{Compare} \in R^4$:

$$A_{Compare} = Comparator(q \mid \hat{a_1} \mid \hat{a_2}) \quad (7)$$

where $\hat{a_1}$ and $\hat{a_2}$ respectively represent the last sub-answer corresponding to the two question subjects, and the four states of the comparison result $A_{Compare} \in R^4$ are respectively represents – "0: not equal, 1: equal, 2: the first option meets, 3: the last option meets".

4 Experiment

4.1 Dataset

We use 2WikiMultiHopQA dataset[2] newly proposed by Ho et al. (2021) to implement the experiments. The 2WikiMultiHopQA dataset contains a total of 192,606 questions jointly constructed through the Wikipedia document set and the Wikidata knowledge base, all of which require multi-hop reasoning. The dataset follows the similar design of HotpotQA (Yang et al., 2018), and the data are split into a training set (167454 questions), a development set (12576 questions) and a test set (12576 questions). All questions in development and test sets are hard multi-hop cases. At the same time, the 2WikiMultiHopQA dataset is also divided into four different question types, namely Compositional, Inference, Comparison and Bridge-comparison.

Compared with HotpotQA (Yang et al., 2018), the 2WikiMultiHopQA dataset removes simple-level questions, increases the types of questions, and the length of the questions and the diversity of answer forms. In addition to following the setting of HotpotQA, Ho et al.(2021) also added the prediction task of the evidence path, which further tested the reasoning and interpretation capabilities of the multi-hop QA model.

The performance evaluation of 2WikiMultiHopQA dataset takes into account the evaluation of the answer, the supporting facts, and the evidence path, using two evaluation metrics: exact match (EM) and F1 score.

4.2 Experimental Details

The Relation Extractor-Reader and Comparator (RERC) model we proposed is divided into three independently trained modules: Relation Extractor, Reader and Comparator.

Relation Extractor uses pre-trained BERT-base model released by Devlin et al. (Devlin et al., 2018) with question length $l = 128$, hidden layer size $d = 768$.

For the CRE model, we collect the relation labels in the given evidence path in the dataset as the classification category labels, a total of 35 categories; for the SRE model, we construct 1,000 samples according to the relation span in the question text through crowdsourcing to train the span extraction pointer network.

Reader uses the ALBERT-large model released by Lan et al. (2020) with $l = 512$ and $d = 1024$, which has been shown advanced performance in the SQuAD 1.1/2.0 dataset (Rajpurkar et al., 2016, 2018).

Comparator use the model structure similar to the CRERC model with $l = 256$ and $d = 768$.

During training, we use the Adam optimizer in all three modules, set the $batch\ size$ to 32,16,32, and the learning rate of 2×10^{-5}, 1×10^{-5}, 2×10^{-5} separately. The learning rate for parameters in BERT warmup over the first 10% steps, and then linearly decays to zero. The hyperparameter of the loss function in RE is set to $\alpha = \beta = 1.0$.

In addition, we also proposed the QETPS method described in the section 3.2. We use the Named Entity Recognition (NER) tool $Stanford\ corenlp\ toolkit$ (Manning et al., 2014) to extract the corresponding named entities from all texts, and then use the threshold of $\sigma_1 = 0.8$ and $\sigma_2 = 0.65$ to match the entity nodes and question relation.

All experiments are based on four Tesla P100 GPUs. In order to determine the proposed method

[2]The dataset benchmark platform located at https://github.com/Alab-NII/2wikimultihop.

	Model	Answer		Sp fact		Evidence		Joint	
		EM	F1	EM	F1	EM	F1	EM	F1
Dev	Ho et al. (2021)	35.30	42.45	23.85	64.31	1.08	14.77	0.37	5.03
	Yang et al. (2018)	34.14	40.95	26.47	66.94	-	-	-	-
	*DFGN (Xiao et al., 2019)	30.87	38.49	17.06	57.79	-	-	-	-
	*QFE (Nishida et al., 2019)	37.56	43.21	21.13	59.20	-	-	-	-
	*QFE + Evidence Extractor	38.30	44.22	34.62	72.18	6.62	33.68	3.57	13.53
	*DecompRC (Min et al., 2019)	7.46	41.57	56.49	82.73	-	-	-	-
	*DecompRC + Comparator	39.94	61.46	68.45	85.54	-	-	-	-
	CRERC	**71.56**	**74.51**	**86.00**	**92.75**	**55.88**	**70.32**	**50.59**	**60.21**
	SRERC	69.74	73.81	81.89	89.95	8.26	25.67	7.66	21.80
Test	Ho et al. (2021)	36.53	43.93	24.99	65.26	1.07	14.94	0.35	5.41
	Human	80.67	82.34	85.33	92.63	57.67	75.63	53.00	66.69
	CRERC	**69.58**	**72.33**	**82.86**	**90.68**	**54.86**	**68.83**	**49.80**	**58.99**

Table 1: Results on the development set and the test set of 2WikiMultiHopQA dataset. The mark * means the models we reproduced according to the open source code and the settings in the original paper. The mark - means those models have no ability to extract the evidence result.

in each stage, we compared a variety of methods through experiments which are described at Appendix A.

4.3 Baseline

We will compare the performance of our RERC model and the previous works on the 2WikiMultiHopQA dataset (Ho et al., 2021).

Ho et al. (2021) The strong baseline model released in the original 2WikiMultiHopQA paper (Ho et al., 2021). It was based on the multi-hop model proposed by Yang et al. (2018), and added a new component to perform the evidence generation task.

DFGN (Xiao et al., 2019) The classic end-to-end multi-hop QA model based on graph neural network, originally working on HotpotQA (Yang et al., 2018) dataset. We reproduced the DFGN model by using the BERT-base pre-trained model (Devlin et al., 2018) under the source code and hyperparameter settings published by Yang et al. (2018).

DecompRC (Min et al., 2019) The classic multi-hop QA model that using question decomposition methods, originally working on HotpotQA (Yang et al., 2018) dataset. We reproduced the DecompRC model by using the same question decomposition method as Min et al. (2019) and the same Reader module as our RERC model, which is helpful to compare our method with the DecompRC model in question decomposition.

QFE (Nishida et al., 2019) The classic multi-hop QA model which was based on the multi-hop model proposed by Yang et al. (2018), and added a Query-Focused Extractor(QFE) module to extract the supporting sentences. We reproduced the QFE model following Nishida et al. (2019).

Human Ho et al. (2021) randomly selected 100 samples in the test set to evaluate human performance.

Next is the introduction of some variants,

CRERC -wo QETPS The CRERC model which does not use the QETPS method but add all paragraphs.

CRERC -w PSBERT The CRERC model which does not use the QETPS method but the paragraph selector of the BERT model applied in DFGN (Xiao et al., 2019).

DecompRC + Comparator The variant of the **DecompRC** model of which the final answer is obtained through the Comparator module proposed in this work.

QFE + Evidence Extractor The variant of the **QFE** model which adds the same Evidence Extractor component as the original baseline model (Ho et al., 2021).

4.4 Results

Table 1 shows the evaluation result of our proposed Relation Extractor-Reader and Comparator (RERC) model on the development set and the test set of 2WikiMultiHopQA dataset (Ho et al., 2021). Our proposed Classification-type Relation Extractor-Reader and Comparator (CRERC) model outperforms all competitors in the evaluation met-

model	Relation Extractor						Reader		Comparator
	question subject			question relation			question type		
	EM	F1	Accuracy	EM	F1	Accuracy	EM	F1	Accuracy
CRERC	0.860	0.955	0.999	-	-	1.000	0.940	0.958	0.976
SRERC	0.860	0.955	-	0.997	0.997	1.000	0.916	0.942	0.976

Table 2: Evaluation of each sub-module in RERC three-stage model. The accuracy of the question relation is only for the CRERC model, while the EM and F1 values only for SRERC model.

Type	Answer		Sp fact		Evidence		Joint	
	EM	F1	EM	F1	EM	F1	EM	F1
Comparison	72.96	73.22	**96.22**	**98.20**	**87.80**	**93.68**	67.11	69.71
Inference	58.30	66.35	73.60	85.18	32.92	46.03	32.73	42.10
Compositional	63.88	68.32	79.74	88.97	36.17	53.38	36.00	47.00
Bridge-Comparison	**92.11**	**92.31**	93.60	98.19	71.07	90.41	**70.16**	**85.06**
All	71.56	74.51	86.00	92.75	55.88	70.32	50.59	60.21

Table 3: CRERC model performance under different question types

rics of answer, support facts and evidences on the development set and the test set. Compared with human performance, our CRERC model is close to human performance in the evaluation metrics of support facts and evidences, with only 1.95 gap in F1 score of support facts. Although the objective indicators of the SRERC model for evidence are low, the evidence path generated by the SRERC model have better readability through human subjective observations, which we will describe in detail in the section 5.3.

In addition to the overall performance evaluation of the model, we also conducted a separate performance evaluation for each part of the three-stage modules. The specific evaluation results are shown in the table 2, where the accuracy of the question relation is only for the CRERC model, and the EM and F1 values are only for SRERC model.

In the table 2, we find that for the Relation Extractor module and the Comparator module, our proposed model has reached very high accuracy, which may be due to the fact that there are a few types of question relations and quantitative relationship comparison in the 2WikiMultiHopQA dataset. The performance of the Reader module has also reached such amazing accuracy as $EM = 0.940$ and $F1 = 0.958$. Therefore, the future research of the question decomposition multi-hop QA model should focus on how to reduce the cumulative error of multiple hops and how to recognize and redress the errors of the previous reasoning steps when performing the next reasoning step.

5 Discussion

5.1 Impact of different problem types

To study the impact of different question types in the 2WikiMultiHopQA dataset, we perform some experiments to compare the CRERC model under each question type, where the results are shown in the table 3. We observed the best performance for our CRERC model in the Bridge-Comparison questions, which combine the Compositional-type and Comparison-type, and have the most number of hops and support facts to to be retrieved, and are designed to be the most challenging question type. We analyzed that it is due to our CRERC model's special method of decomposing complex questions based on relation extraction, which is not interfered by the expression of compound question types. Besides we find the question relation setting of Bridge-Comparison questions is relatively simple, and the sub-question is easier to answer, which offset the impact of more hops.

In general, the RERC model performs significantly better on Comparison-type and Bridge-Comparison-type than Compositional-type and Inference-type, which is due to that the Comparison-type and Bridge-Comparison-type questions have easier sub-questions, as compensa-

Model	Answer		Sp fact		Evidence		Joint	
	EM	F1	EM	F1	EM	F1	EM	F1
CRERC	**71.56**	**74.51**	**86.00**	**92.75**	**55.88**	**70.32**	**50.59**	**60.21**
CRERC -wo QETPS	37.13	38.79	20.89	54.34	6.63	16.72	0.07	2.27
CRERC -w PSBERT	68.77	71.77	81.54	88.27	53.64	67.12	46.27	55.67

Table 4: Results of Ablation experiment about QETPS method

tion for additional comparison tasks, which can be accomplished greatly by our Comparator module.

Model	manual scoring
CRERC	4.03 ± 0.58
SRERC	4.22 ± 0.52

Table 5: Manual evaluation of evidence path

5.2 Impact of QETPS

Due to the length limitation of text the Reader module can process one time and the large number and long lengths of context in the dataset, we designed a Query-aware Entity Tree Paragraph Screening (QETPS) method to filter these paragraphs. In order to verify whether the QETPS method we introduced is effective, we executed ablation experiments to compare the performance changes after replacing the QETPS method with the BERT-based paragraph selector used in DFGN model (Xiao et al., 2019). The results of the ablation experiment are shown in the table 4.

In the table 4, we find that without using any paragraph filtering method, the Reader is likely to be unable to find the answer to the sub-question from messy paragraphs, resulting in a significant performance degradation. Compared with the results of using the BERT-based paragraph selector in the DFGN model (Xiao et al., 2019), our QETPS method has achieved better performance, which may be due to our QETPS method makes good use of the entity information in the paragraph, which is just the hop intermediary in multi-hop QA tasks.

5.3 Results of Evidence Path Generation : Manual Evaluation

Previously in the table 1, we found that the SRERC model did not perform well in the evidence path metric. However, we analyzed that the unsatisfactory performance is due to that the evidences in the 2WikiMultiHopQA dataset are derived from the tags of the Wikidata knowledge base, which may not appear in the text of question and context. Our SRERC model uses the fragments in the question as the relation in the evidence path, which results in lower score on objective indicators.

We believe that the evidences of the multi-hop QA model should be expressed in free style, which is difficult to evaluate with objective indicators. As the result, we re-evaluated it through manual evaluation. We randomly selected 100 samples from every question-types to show the evidence path and final predictions of the CRERC model and the SRERC model[3]. Each samples was scored by seven graduate students for the evidence extraction capabilities of the two models. We use a score of 1 to 5 to indicate whether the worker believes that the model faithfully demonstrated its reasoning process and got the correct answer. The table 5 shows the results of manual evaluation. We can surprisingly discover that the SRERC model has obtained a higher manual score than the CRERC model. We guess the reason that the expression from the question fragment is easier to reveal the reasoning process of the model. Of course, our conclusions may be biased due to the bias of workers. Therefore, we will continue to explore more rigorous evaluation method for evidence path in our future work.

6 Conclusion and future work

We propose a three-stage framework of Relation Extractor-Reader and Comparator (RERC), which solves the multi-hop QA task through the idea of complex question decomposition, and obtains the state-of-the-art results in the 2WikiMultiHopQA dataset, which is close to human performance. Our RERC framework can also provide faithful evidence with excellent interpretability.

Multiple future research directions according to our proposed RERC model may be envisioned.

[3]Some cases are shown in the appendix B.

First of all, benefiting to the three-stage structure, the RERC model has the potential to leverage the network structure of the Relation Extractor to gain generalization capabilities for more complex questions. Moreover, we expect that collaborative error correction mechanism applied in Reader module will largely avoid accumulation of errors in multi-hop reasoning.

Acknowledgements

The work is supported by The Youth Innovation Promotion Association of the Chinese Academy of Sciences (E1291902), Jun Zhou (2021025). We would like to thank Jiahao Yang, Ming Zhang, Jianzhong Kuang and Chengzhang Li for their valuable support in the procedure of Manual Evaluation. We thank the responsible reviewers for their insightful feedback and valuable suggestions.

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. In *Proceedings of the 2019 Annual Meeting of the International Conference on Learning Representations(ICLR)*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *Computer Science*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Bhuwan Dhingra, Qiao Jin, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2018. Neural models for reasoning over multiple mentions using coreference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational*.

Ming Ding, Chang Zhou, Qibin Chen, Hongxia Yang, and Jie Tang. 2019. Cognitive graph for multi-hop reading comprehension at scale. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2021. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of COLING 2021*.

Naoya Inoue, Pontus Stenetorp, and Kentaro Inui. 2020. R4C: A benchmark for evaluating rc systems to get the right answer for the right reason. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *Proceedings of the 2020 Annual Meeting of the International Conference on Learning Representations(ICLR)*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*.

Sewon Min, Victor Zhong, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2019. Multi-hop reading comprehension through question decomposition and rescoring. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Kosuke Nishida, Kyosuke Nishida, Masaaki Nagata, Atsushi Otsuka, Itsumi Saito, Hisako Asano, and Junji Tomita. 2019. Answering while summarizing: Multi-task learning for multi-hop QA with evidence extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ethan Perez, Patrick Lewis, Wen-tau Yih, Kyunghyun Cho, and Douwe Kiela. 2020. Unsupervised question decomposition for question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

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*.

Linfeng Song, Zhiguo Wang, Mo Yu, Yue Zhang, Radu Florian, and Daniel Gildea. 2018. Exploring graph-structured passage representation for multi-hop reading comprehension with graph neural networks. *arXiv preprint arXiv:1809.02040*.

Alon Talmor and Jonathan Berant. 2018. The web as a knowledge-base for answering complex questions. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computationa*.

Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. 2019. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *Proceedings of the 2018 Annual Meeting of the International Conference on Learning Representations(ICLR)*.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. *Computer Science*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*.

Yunxuan Xiao, Yanru Qu, Lin Qiu, 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*.

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*.

Victor Zhong, Caiming Xiong, Nitish Shirish Keskar, and Richard Socher. 2019. Coarse-grain fine-grain coattention network for multi-evidence question answering. In *Proceedings of the 2019 Annual Meeting of the International Conference on Learning Representations(ICLR)*.

Appendix

A Alternative Methods in Each Stage

In this section, We evaluated several methods in each stage according to the task characteristics which is briefly mentioned in Section 3 due to page limit and chose the best one. We show the performance comparison and analysis of those alternative methods in table 6 and table 7.

Model	Subject	Relations		Type
	F1	Acc.	F1	Acc.
BERT	0.947	0.995	0.976	1.000
BERT+Transformer	0.955	0.994	0.986	1.000
BERT+Type Fuse	0.947	0.999	0.994	1.000
BERT+Transformer+Type Fuse	**0.955**	**0.999**	**0.997**	**1.000**

Table 6: The evaluation results of alternative methods for Relation Extractor module. Note that in the Relations area the Acc. is for CRERC model and the F1 value is for SRERC model.

Model	**CRERC**		**SRERC**	
	EM	F1	EM	F1
BiDAF	0.679	0.713	0.661	0.709
BERT base	0.835	0.862	0.803	0.841
BERT large	0.867	0.895	0.832	0.846
Roberta base	0.916	0.930	0.884	0.924
Roberta large	0.922	0.944	0.895	0.921
ALBERT large	**0.940**	**0.958**	0.916	**0.942**
ALBERT xlarge	0.932	0.952	**0.920**	0.932

Table 7: The evaluation results of alternative methods for Reader module.

B Output Cases

In this section, we show some cases of CRERC model and SRERC model output for the 2WikiMultiHopQA dataset, including question text, gold label, RE module output (type, subject, relation) and final output (evidence path, prediction) .

- **Case 1**

  ID: 161093c40bde11eba7f7acde48001122

  Question: What is the place of birth of Kévin Ledanois's father?

  Gold Label: Montreuil

  Type: compositional

  Subject: Kévin Ledanois

  ——-CRERC Predict——-

  Relation: father; place of birth

  Evidence: (Kévin Ledanois; father; Yvon Ledanois) ⇒ (Yvon Ledanois; place of birth; Montreuil-sous-Bois))

  Predict: Montreuil-sous-Bois)

  ——-SRERC Predict——-

  Relation: father; place of birth

  Evidence: (Kévin Ledanois; father; Yvon Ledanois) ⇒ (Yvon Ledanois; place of birth; Montreuil-sous-Bois))

Predict: Montreuil-sous-Bois)

- **Case 2**

  ID: 17ba791a0bde11eba7f7acde48001122

  Question: What nationality is the director of film Top Floor Girl?

  Gold Label: French

  Type: compositional

  Subject: Top Floor Girl

  ——-CRERC Predict——-

  Relation: director; country of citizenship

  Evidence: (Top Floor Girl; director; Max Varnel) ⇒ (Max Varnel; country of citizenship; French-born)

  Predict: French-born

  ——-SRERC Predict——-

  Relation: director; nationality

  Evidence: (Top Floor Girl; director; Max Varnel) ⇒ (Max Varnel; nationality; French-born)

  Predict: French-born

- **Case 3**

  ID: 8f038cdb096011ebbdafac1f6bf848b6

  Question: Which film came out earlier, Aram + Aram = Kinnaram or Thayagam?

  Gold Label: Aram + Aram = Kinnaram

  Type: comparison

  Subject: Aram + Aram = Kinnaram; Thayagam

  ——-CRERC Predict——-

  Relation: publication date

  Evidence: (Aram + Aram = Kinnaram; publication date; 1985)

  (Thayagam; publication date; 1996)

  Predict: Aram + Aram = Kinnaram

  ——-SRERC Predict——-

  Relation: came out

  Evidence: (Aram + Aram = Kinnaram; came out; 1985)

  (Thayagam; came out; 1996)

  Predict: Aram + Aram = Kinnaram

- **Case 4**

  ID: 17e3349208df11ebbd9fac1f6bf848b6

  Question: Who is younger, Osita Chidoka or David Faurschou?

  Gold Label: Osita Chidoka

  Type: comparison

  Subject: David Faurschou; Osita Chidoka

  ——-CRERC Predict——-

  Relation: date of birth

  Evidence: (David Faurschou; date of birth; January 28, 1956))

  (Osita Chidoka; date of birth; 18 July 1971))

  Predict: Osita Chidoka

  ——-SRERC Predict——-

  Relation: younger

  Evidence: (David Faurschou; younger; January 28, 1956))

  (Osita Chidoka; younger; 18 July 1971))

  Predict: Osita Chidoka

- **Case 5**

  ID: 8762e83a0baf11ebab90acde48001122

  Question: Who is the paternal grandfather of Kerry Earnhardt?

  Gold Label: Ralph Earnhardt

  Type: inference

  Subject: Kerry Earnhardt

  ——-CRERC Predict——-

  Relation: father; father

  Evidence: (Kerry Earnhardt; father; Dale Earnhardt) ⇒ (Dale Earnhardt; father; Ralph Earnhardt)

  Predict: Ralph Earnhardt

  ——-SRERC Predict——-

  Relation: grandfather; grandfather

  Evidence: (Kerry Earnhardt; grandfather; Dale Earnhardt) ⇒ (Dale Earnhardt; grandfather; Ralph Earnhardt)

  Predict: Ralph Earnhardt

- **Case 6**

  ID: 6a0a17b80baf11ebab90acde48001122

  Question: Who is Alice Claypoole Vanderbilt's mother-in-law?

  Gold Label: Maria Louisa Kissam

  Type: inference

  Subject: Alice Claypoole Vanderbilt

  ——-CRERC Predict——-

  Relation: spouse; mother

  Evidence: (Alice Claypoole Vanderbilt; spouse; Cornelius Vanderbilt II) ⇒ (Cornelius Vanderbilt II; mother; Maria Louisa Kissam.)

  Predict: Maria Louisa Kissam.

  ——-SRERC Predict——-

  Relation: [CLS]; mother

  Evidence: (Alice Claypoole Vanderbilt; [CLS]; Cornelius Vanderbilt II) ⇒ (Cornelius Vanderbilt II; mother; Maria Louisa Kissam.)

  Predict: Maria Louisa Kissam.

- **Case 7**

  ID: 6bc3222c086511ebbd5eac1f6bf848b6

  Question: Which film has the director who is older, The Woman Next Door or La Estatua De Carne?

  Gold Label: La Estatua De Carne

  Type: bridge comparison

  Subject: La estatua de carne; The Woman Next Door

  ——-CRERC Predict——-

  Relation: director; date of birth

  Evidence: (La estatua de carne; director; Chano Urueta) ⇒ (Chano Urueta; date of birth; February 24, 1904)

  (The Woman Next Door; director; François Truffaut) ⇒ (François Truffaut; date of birth; (6 February 1932)

  Predict: La estatua de carne

  ——-SRERC Predict——-

  Relation: director; older

  Evidence: (La estatua de carne; director; Chano Urueta) ⇒ (Chano Urueta; older; February 24, 1904)

  (The Woman Next Door; director; François Truffaut) ⇒ (François Truffaut; older; (6 February 1932)

  Predict: La estatua de carne

- **Case 8**

  ID: 09646113087011ebbd62ac1f6bf848b6

  Question: Which film has the director died later, Fugitives For A Night or Chinese In Paris?

  Gold Label: Chinese In Paris

  Type: bridge comparison

  Subject: Fugitives for a Night; Chinese in Paris

  ——-CRERC Predict——-

  Relation: director; date of death

  Evidence: (Fugitives for a Night; director; Leslie Goodwins) ⇒ (Leslie Goodwins; date of death; 8 January 1969))

  (Chinese in Paris; director; Jean Yanne) ⇒ (Jean Yanne; date of death; 23 May 2003))

  Predict: Chinese in Paris

  ——-SRERC Predict——-

  Relation: director; die

  Evidence: (Fugitives for a Night; director; Leslie Goodwins) ⇒ (Leslie Goodwins; die; 8 January 1969))

  (Chinese in Paris; director; Jean Yanne) ⇒ (Jean Yanne; die; 23 May 2003))

  Predict: Chinese in Paris