Explainable Multi-hop Question Generation: An End-to-End Approach without Intermediate Question Labeling

Seonjeong Hwang[1], **Yunsu Kim**[3], **Gary Geunbae Lee**[1,2]
[1]Graduate School of Artificial Intelligence, POSTECH, Republic of Korea
[2]Department of Computer Science and Engineering, POSTECH, Republic of Korea
[3]aiXplain, Inc. Los Gatos, CA, USA
seonjeongh@postech.ac.kr, yunsu.kim@aixplain.com, gblee@postech.ac.kr

Abstract

In response to the increasing use of interactive artificial intelligence, the demand for the capacity to handle complex questions has increased. Multi-hop question generation aims to generate complex questions that requires multi-step reasoning over several documents. Previous studies have predominantly utilized end-to-end models, wherein questions are decoded based on the representation of context documents. However, these approaches lack the ability to explain the reasoning process behind the generated multi-hop questions. Additionally, the question rewriting approach, which incrementally increases the question complexity, also has limitations due to the requirement of labeling data for intermediate-stage questions. In this paper, we introduce an end-to-end question rewriting model that increases question complexity through sequential rewriting. The proposed model has the advantage of training with only the final multi-hop questions, without intermediate questions. Experimental results demonstrate the effectiveness of our model in generating complex questions, particularly 3- and 4-hop questions, which are appropriately paired with input answers. We also prove that our model logically and incrementally increases the complexity of questions, and the generated multi-hop questions are also beneficial for training question answering models.

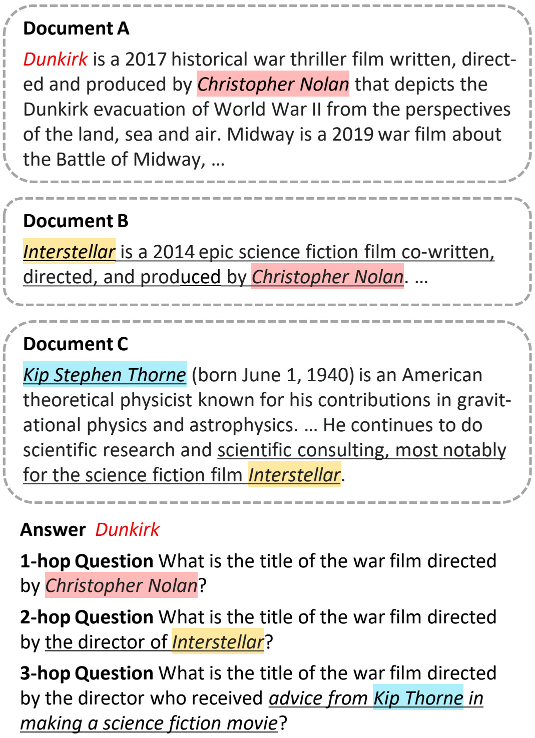
Keywords: Multi-hop Question Generation, Automatic Question Generation

1. Introduction

Question generation (QG) aims to generate questions related to the given context documents, with applications in various domains, including developing chatbots and educational tutoring systems and enhancing datasets used in question answering (QA) models. Recently, with advanced interactive artificial intelligence and chatbots, user questions have become complex, covering a wide range of information. Thus, the demand for machines capable of handling questions that necessitate intricate reasoning and contextual understanding has increased.

In early research, the focus was primarily on single-hop QG generating questions based on a single document or sentence (Du et al., 2017; Zhao et al., 2018; Sun et al., 2018; Chan and Fan, 2019; Alberti et al., 2019). For example, the 1-hop question in Figure 1 is constructed solely from the content of *Document A*. In real-world scenarios, however, we may not be able to recall direct information related to the question we want to ask. Instead, we use additional information that indirectly describes the subject, as indicated in the 2- and 3-hop questions in the figure.

Multi-hop QG involves aggregating the related information scattered across multiple documents and generating questions that require multi-step reasoning. In most previous work, graph-to-sequence (graph2seq) architectures are primarily used to extract meaningful relationships from several contexts and then generate multi-hop questions based on the context representation (Pan et al., 2020; Su



Document A
Dunkirk is a 2017 historical war thriller film written, directed and produced by *Christopher Nolan* that depicts the Dunkirk evacuation of World War II from the perspectives of the land, sea and air. Midway is a 2019 war film about the Battle of Midway, …

Document B
Interstellar is a 2014 epic science fiction film co-written, directed, and produced by *Christopher Nolan*. …

Document C
Kip Stephen Thorne (born June 1, 1940) is an American theoretical physicist known for his contributions in gravitational physics and astrophysics. … He continues to do scientific research and scientific consulting, most notably for the science fiction film *Interstellar*.

Answer *Dunkirk*

1-hop Question What is the title of the war film directed by *Christopher Nolan*?

2-hop Question What is the title of the war film directed by the director of *Interstellar*?

3-hop Question What is the title of the war film directed by the director who received *advice from Kip Thorne in making a science fiction movie*?

Figure 1: Example of multi-hop question generation through question rewriting.

20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

et al., 2020; Fei et al., 2022). However, as the number of referenced documents increases, these end-to-end approaches face limitations in generating logically coherent multi-hop questions. Cheng et al. (2021) proposed a step-by-step question rewriting framework to address this problem. This framework consists of a single-hop QG model and a question rewriting model, which increases the question complexity by leveraging additional information. To train the question rewriting model, the authors labeled intermediate questions that compose 2-hop questions. Inspired by the method of Cheng et al. (2021), we explore the step-by-step question rewriting approach, but it does not require intermediate question labeling.

In this paper, we introduce an End-to-End Question Rewriting (E2EQR) model, which initially generates single-hop questions and then sequentially rewrites them to increase the complexity. The E2EQR model is a type of recurrent neural network (RNN) with a sequence-to-sequence (seq2seq) Transformer (Vaswani et al., 2017) as its backbone. In each step, the model takes a document and uses it to rewrite the question generated in the previous step. Rather than using the generated question as the input for the subsequent steps, we implicitly transfer the hidden states computed during the decoding process in the prior steps, enabling end-to-end training without the ground truth for intermediate questions. Additionally, we design a curriculum learning algorithm that allows the model to learn low- to high-complexity examples sequentially while preventing catastrophic forgetting for lower-hop questions.

In experiments, we conducted both quantitative and qualitative evaluations on the multi-hop questions generated by E2EQR. According to the results, our method achieves performance comparable to the state-of-the-art model, even while performing the additional task of intermediate question generation. Moreover, we prove the efficacy of the question rewriting approach in logically crafting complex questions that accurately align with the input answer. We also observe that this precise capability in generating multi-hop QA pairs significantly contributes to data augmentation for training QA models.

Our contributions can be summarized as follows:

- We introduce an explainable multi-hop QG model that performs step-by-step question rewriting to increase the question complexity[1].
- Our model is trained end-to-end without the need to label intermediate questions.
- Our question rewriting approach demonstrates effectiveness in logically generating complex questions that correspond to input answers, particularly in 3- and 4-hop QG scenarios.
- The synthetic multi-hop QA data also prove to be beneficial for training QA models.

2. Related Work

Early studies on QG have primarily focused on generating questions based on their syntactic and semantic patterns (Wolfe, 1976; Heilman and Smith, 2010; Heilman, 2011; Mazidi and Nielsen, 2014). However, these approaches were limited to generating factual questions from short sentences or paragraphs. The seq2seq neural network has alleviated these limitations and enabled generating diverse types of questions. Recent work has proposed various approaches using attention-based seq2seq models (Du et al., 2017; Zhao et al., 2018; Sun et al., 2018) or pretrained language models (Chan and Fan, 2019; Alberti et al., 2019), which encode input documents and answers, and then generate related questions. These studies aimed to generate single-hop questions using single-hop QA datasets, such as SQuAD (Rajpurkar et al., 2016), NewsQA (Trischler et al., 2016), Natural Questions (Kwiatkowski et al., 2019).

Multi-hop QG requires a comprehensive understanding of the content scattered across multiple documents. Pan et al. (2020) and Su et al. (2020) encoded input documents using graph neural networks (GNNs) to capture relationships between scattered information. Fei et al. (2022) proposed a hard-controlled generation framework that ensures multi-hop QG by forcing the decoding process to use key entities extracted from the input documents. Wang et al. (2020) and Xie et al. (2020) employed reinforcement learning to reflect the fluency, relevance or correspondence to the input answers of the generated questions for model training. Pan et al. (2021) proposed an unsupervised method that generates single-hop questions and fuses them into multi-hop questions. Cheng et al. (2021) proposed a step-by-step question rewriting framework to logically control the question difficulty. The authors decomposed 2-hop questions into single-hop intermediate questions and trained two generation models: a single-hop QG model and a question rewriting model. In this study, inspired by the approach of Cheng et al. (2021), we develop an E2EQR model that can be trained without labeled intermediate questions.

3. Method

The E2EQR model is a Transformer-based RNN that generates questions based on the current en-

[1] We release our code on https://github.com/SeonjeongHwang/e2eQR

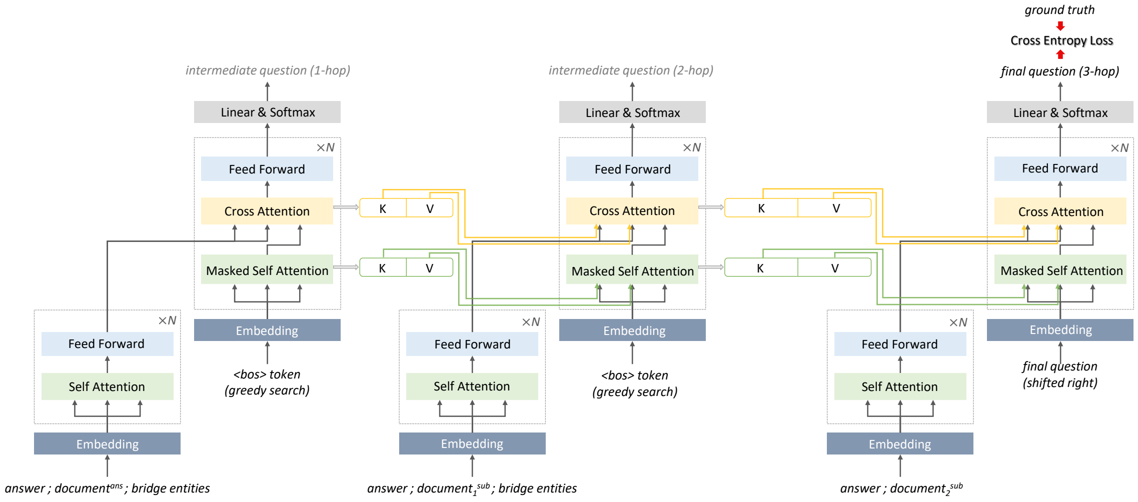


Figure 2: Unfolded architecture of the proposed model. Training process for 3-hop question generation. In the decoder, the multi-head masked self-attention and multi-head cross attention layers use the key and value matrices (K and V) accumulated from the prior steps to rewrite the intermediate question generated in the previous step. We omitted the detailed elements of the Transformer (Vaswani et al., 2017) in this figure.

coder input and hidden states computed while decoding in the previous steps. As illustrated in Figure 2, the model sequentially processes input elements and either generates the initial question or rewrites the previous one. In the first step, the document that contains the answer is input into the model with the answer and bridge entities. Then, the model generates a 1-hop intermediate question using a greedy search. The bridge entities carry the contents of future input documents, facilitating the generation of questions that can be rewritten. In the tth step ($t > 1$), the model rewrites the $(t-1)$-hop question into the t-hop question. Rather than explicitly taking the question generated in the previous step as input, our model relies on accumulated decoder hidden states to rewrite it. This allows the model to be trained using only the ground truth of the final $\mathcal{N}$-hop questions.

3.1. Problem Formulation

Given the $\mathcal{N}$ document set $\{\mathcal{C}\}_{1:\mathcal{N}}$ and the target answer $\mathcal{A}$, the objective is to generate the $\mathcal{N}$-hop question $\mathcal{Q}^{\mathcal{N}}$ constructed based on inferences drawn from all given documents. The document containing the answer and the rest of the documents are denoted as *document*^{ans} ($\mathcal{C}^{ans}$) and *document*^{sub} ($\mathcal{C}^{sub}$), respectively. The tth document $\mathcal{C}_t$ and t-hop question $\mathcal{Q}^t$ are the sequence of l_t and m_t tokens, respectively (i.e., $\mathcal{C}_t = c_{1:l_t}$ and $\mathcal{Q}^t = q_{1:m_t}$).

3.2. Document Arrangement and Bridge Entity Extraction

In Figure 1, the entities *“Christopher Nolan”* and *“Interstellar”* are the bridges for *Document* <*A* to *B*> and *Document* <*B* to *C*>. In addition, *Document B* and *C* are sequentially referenced to rewrite the 1-hop question as a 3-hop question. As the example, the input documents should be input to the model in an order that allows sequential rewriting. In addition, the model should know which entities will appear again in future input documents. To extract bridge entities and arrange the input documents, we construct a document graph.

First, we extract bridge entities that commonly appear in more than two documents using named entity recognition and key phrase extraction tools[2][3]. Next, we construct a document graph where the nodes represent input documents, and two nodes sharing bridge entities are connected by an edge. Then we serialize the graph through a breadth-first-search with *document*^{ans} as a root. The arranged documents are sequentially input to each step of E2EQR along with the answer and their bridge entities.

3.3. End-to-End Question Rewriting

Given $\mathcal{N}$ documents, bridge entities, and the answer, E2EQR aims to generate an optimal $\mathcal{N}$-hop

[2] https://demo.allennlp.org
[3] https://spacy.io

question:

$$\mathcal{Q}^{\mathcal{N}*} = \arg\max_{\mathcal{Q}^{\mathcal{N}}} P(\mathcal{Q}^{\mathcal{N}} \mid \{\mathcal{C}\}_{1:\mathcal{N}}, \{\mathcal{B}\}_{1:(\mathcal{N}-1)}, \mathcal{A}; \theta),$$

where $\mathcal{B}_t$ represents a set of bridge entities input with $\mathcal{C}_t$, and θ denotes the model parameters, consisting of the encoder and decoder parameters: θ_{enc} and θ_{dec}. The model does not take the bridge entities in the $\mathcal{N}$th step, and the model parameters are shared in all steps.

In the first step, the encoder takes the input sequence and returns the output representation $H_1 \in R^{l_1 \times dim_{model}}$:

$$H_1 = \text{Encoder}(\mathcal{C}_1, \mathcal{B}_1, \mathcal{A}; \theta_{enc}),$$

where l_1 represents the length of the input sequence, and dim_{model} denotes the output dimension. With the encoder output representation, the decoder performs autoregressive decoding from the <*bos*> token until the <*eos*> token is returned:

$$P(\mathcal{Q}^1 \mid H_1; \theta_{dec}) = \prod_{i=1}^{m_1} P(q_i \mid q_{1:i-1}, H_1; \theta_{dec}),$$

where the length of the 1-hop question is m_1.

In the tth ($t > 1$) step, the encoder output representation $H_t \in R^{l_t \times dim_{model}}$ is obtained in the same way as the first step. However, from the tth step, the decoder uses the encoder output representation and the decoder hidden states calculated in the prior steps.

The Transformer decoder contains the multi-head masked self-attention (SA) layers and multi-head cross-attention (CA) layers[4], and the layer output is the attention value computed from query Q, key K, and value V matrices (Vaswani et al., 2017). The SA mechanism employs the projections of the decoder input representation as $Q \in R^{m'_t \times d_k}$, $K \in R^{m'_t \times d_k}$, and $V \in R^{m'_t \times d_k}$ matrices, where m'_t ($1 \leq m'_t \leq m_t$) is the decoder input length, and d_k is the attention hidden dimension. In the CA layer, the key and value matrices are the projections of the encoder output representation H_t (i.e., $K \in R^{l_t \times d_k}$ and $V \in R^{l_t \times d_k}$).

In the proposed method, we introduce the **accumulated SA** and **accumulated CA** mechanisms, which employ the accumulated key and value matrices, $K_{1:t}$ and $Q_{1:t}$, the concatenations of matrices computed in the prior steps and the current tth step. In other words, the scaled dot-product attention function in the accumulated attention layers is as follows:

$$\text{Attention}(Q_t, K_{1:t}, V_{1:t}) = \text{Softmax}(\frac{Q_t K_{1:t}^T}{\sqrt{d_k}}) V_{1:t}.$$

The size of the matrices $K_{1:t}$ and $V_{1:t}$ is $(\sum_{i=1}^{i=t-1} m_i + m'_t) \times d_k$ in the accumulated SA and $(\sum_{i=1}^{i=t} l_t) \times d_k$ in the accumulated CA. In summary, the question decoding in the tth step is as follows:

$$\begin{aligned} & P(\mathcal{Q}^t \mid H_t, K_{1:t-1}, V_{1:t-1}; \theta_{dec}) \\ & = \prod_{i=1}^{m_t} P(q_i \mid q_{1:i-1}, H_t, K_{1:t-1}, V_{1:t-1}; \theta_{dec}). \end{aligned}$$

The accumulated SA and CA mechanisms allow the use of the information from the intermediate questions and input documents of the prior steps, respectively. Moreover, instead of using the question generated in the previous step as input to the next step, our model performs implicit question rewriting via these methods. Consequently, this method allows end-to-end training of the model without the ground truth of the intermediate questions. We train our model using the cross-entropy loss for the question generated in the final step.

3.4. Adaptive Curriculum Learning

The proposed model generates $\mathcal{N}$-hop questions based on the capability of generating 1- to $(\mathcal{N}-1)$-hop questions. Therefore, we design a curriculum learning algorithm where the model learns sequentially from low-hop questions to high-hop questions. However, we empirically observed that sequential learning by complexity level causes catastrophic forgetting. Thus, we apply intensive multitask learning, where the weight assigned to each complexity changes as the training progresses.

Algorithm 1 Adaptive Curriculum Learning

```
Input: $\{D_1, D_2, \cdots, D_{\mathcal{N}}\}$, E2EQR$_\theta$, $\alpha$, $\gamma_{low}$, $\gamma_{high}$, $\rho$
for $\mathcal{H} = 1$ to $\mathcal{N}$ do
    Training examples $\mathfrak{D} \leftarrow D_1 \cup \cdots \cup D_{\mathcal{H}}$
    for $h = \mathcal{H} + 1$ to $\mathcal{N}$ do
        $\mathfrak{D} \leftarrow \mathfrak{D} \cup D_h^{part}$, where $n(D_h^{part}) = \rho \cdot n(D_h)$
    end for
    for $i = 1$ to $n(\mathfrak{D})$ do
        $x_i, y_i \leftarrow \mathfrak{D}_i$
        $\mathcal{L}_i \leftarrow$ CrossEntropyLoss($y_i$, E2EQR$_\theta(x_i)$)
        if $i < \mathcal{H}$ then
            $\mathcal{L}_i \leftarrow \gamma_{low} \cdot \mathcal{L}_i$
        end if
        if $i > \mathcal{H}$ then
            $\mathcal{L}_i \leftarrow \gamma_{high} \cdot \mathcal{L}_i$
        end if
    end for
    $\mathcal{L} \leftarrow \sum_{i=1}^{n(\mathfrak{D})} \mathcal{L}_i / n(\mathfrak{D})$
    $\theta \leftarrow \theta - \alpha \cdot \frac{\partial \mathcal{L}}{\partial \theta}$
end for
```

Algorithm 1 describes the overall process of training E2EQR$_\theta$ with examples $\{D_1, D_2, \cdots, D_{\mathcal{N}}\}$ grouped by the question complexity. In this algorithm, the main complexity $\mathcal{H}$ is assigned to each iteration, which increases from 1 to $\mathcal{N}$, and the

[4] For a simple description, we omitted concepts regarding Transformer multi-head attention.

rest are treated as subcomplexities (Line 1). In the training iteration with the main complexity $\mathcal{H}$, all examples from D_1 to $D_{\mathcal{H}}$ are used as training examples $\mathfrak{D}$ (Line 2). The examples from D_1 to $D_{\mathcal{H}-1}$ are used to prevent catastrophic forgetting. In addition, examples D_h^{part} with a higher complexity than $\mathcal{H}$ are also used to prevent overfitting for lower-hop questions, and the number of examples is controlled using the suppression ratio ρ (Lines 3–5).

Lines 6–15 display the process of updating model parameters θ using the examples $\mathfrak{D}$ and the learning rate α. At this stage, we employ two hyperparameters: the loss weight for lower complexity levels γ_{low} and the loss weight for higher complexity levels γ_{high}. These loss weights allows the model to focus on training examples of the main complexity. We use different loss weights for the two groups because the model already learns the examples of lower complexities. The loss weight is applied to the computed loss according to the data complexity. This algorithm represents a simplified version, and in actual experiments, we switched the main complexity after optimizing the model for that specific complexity. In addition, we used mini-batch stochastic gradient descent and learning rate scheduling, which linearly decreases the learning rate.

4. Experiments

4.1. Dataset

We used two multi-hop QA datasets, MuSiQue (Trivedi et al., 2022) and HotpotQA (Yang et al., 2018), to compare our model with the baselines. HotpotQA contains crowd-sourced 1- and 2-hop questions that require reasoning on supporting facts collected from Wikipedia. MuSiQue is a more challenging dataset, containing up to 4-hop questions constructed by compositing 1-hop questions. The authors of this dataset aimed to address certain shortcomings in HotpotQA, where answers can be derived by reasoning from only partial supporting evidence. In the experiments, we mainly used MusiQue to verify that our model robustly generates questions of varying complexity levels.

The test sets for both datasets are not publicly available; thus, we used the examples from the part of examples from the original training set as the validation set and the original validation set as the test set. Therefore, we used 25,483/700/2,417 examples and 89,947/500/7,405 examples for training, validation and test on MuSiQue and HotpotQA, respectively. Additionally, we split the MuSiQue training set into 12,742 $seen$ and 12,741 $unseen$ examples for generating a synthetic training set in Section 6.1.

4.2. Metric

We measured BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and ROUGE (Lin, 2004) scores using `pycocoevalcap`[5]. All metrics evaluate the n-gram similarity between the prediction and ground truth. The BLEU score measures n-gram word precision and we specifically used the 4-gram-based BLEU (BLEU-4). The METEOR score calculates the F1 score for explicit word-to-word alignment, and ROUGE evaluates the n-gram similarity using the F1 score. We used ROUGE-L, focusing on the longest common subsequence. However, it is insufficient to evaluate the question quality based only on the similarities to reference questions. Therefore, we conducted a qualitative analysis through human evaluation.

4.3. Baselines

We compared our model with following strong multi-hop QG baselines[6]:

- **DP-Graph** (Pan et al., 2020) is a graph2seq model that generates questions based on the word-level document representation and node-level semantic graph representation encoded using an attention-based GNN.
- **CQG** (Fei et al., 2022) generates multi-hop questions using a hard-controlled generator, which controls the decoding process so that the key entities extracted from documents are included in the generated question.
- **MulQG** (Su et al., 2020) is a graph2seq model using a graph convolution network to encode the input documents in consideration of the answer.
- **BART** (Lewis et al., 2020) is a seq2seq Transformer pretrained with the text denoising task. We used the *large* model and trained the model to generate multi-hop questions given the concatenation of the answer and documents.

We also examined whether a large language model can perform multi-hop QG through in-context learning. The results are reported in Appendix A.

4.4. Implementation Details

We initialized our model parameters using `facebook/BART-large` released on Hugging Face[7]. We used AdamW (Loshchilov and Hutter, 2017) optimizer with a batch size of 8, a learning

[5] https://github.com/salaniz/pycocoevalcap

[6] We used GitHub source codes released by the authors to train the baselines on MuSiQue.

[7] https://huggingface.co

Model	2-hop			3-hop			4-hop			Intermediate
	BLEU-4	METEOR	ROUGE-L	BLEU-4	METEOR	ROUGE-L	BLEU-4	METEOR	ROUGE-L	Question
DP-Graph (Pan et al., 2020)	5.14	10.80	28.69	4.33	10.37	28.33	4.47	9.87	28.01	
CQG (Fei et al., 2022)	9.64	16.20	33.98	7.79	13.77	31.12	5.14	11.93	26.48	
MulQG (Su et al., 2020)	9.56	15.41	37.18	9.23	14.35	35.66	7.13	12.43	31.88	
BART (Lewis et al., 2020)	20.84	25.91	43.81	17.64	22.93	41.16	16.11	20.63	37.02	
E2EQR	20.33	25.64	44.01	17.02	22.33	40.04	15.34	19.78	36.98	✓

Table 1: Automatic evaluation results on the MusiQue test set.

Model	2-hop			3-hop			4-hop		
	Fluency	Complexity (≤ 2)	Answer Matching	Fluency	Complexity (≤ 3)	Answer Matching	Fluency	Complexity (≤ 4)	Answer Matching
BART	4.80	1.93	87.5%	4.88	2.54	75.0%	4.78	2.92	73.8%
E2EQR	4.92	1.99	87.5%	4.83	2.50	82.5%	4.60	2.96	80.0%
Ground Truth	4.85	1.99	95.0%	4.85	2.65	78.8%	4.65	3.29	77.5%

Table 2: Human evaluation of questions generated by multi-hop QG models and the ground truth.

rate of 3e-5, 1000 learning rate warm-up steps. The hyperparameters ρ, γ_{low}, and γ_{high} were set to 0.1, 0.8, and 0.1, which is the optimal combination on the validation set and searched from {1, 0.1, 0.01}, {1, 0.8, 0.5} and {0.1, 0.01}, respectively. Our models and baselines were trained using five random seeds and we reported the average performance in Section 5.1.

5. Results

5.1. Automatic Evaluation

Table 1 summarizes the performance of E2EQR and baseline models on the MuSiQue test set. According to the results, our model significantly surpasses DP-Graph, CQG, and MulQG across data of all complexity levels. These models are optimized for the HotpotQA dataset, which comprises 2-hop questions with shorter supporting evidence compared to MuSiQue. We attribute the poor performance of these models on MuSiQue to this disparity in data characteristics. E2EQR and BART exhibit comparable results, as they share the same backbone. However, our model, despite having the same number of parameters as BART, generates multi-hop questions as well as their intermediate questions. We also report the results on HotpotQA in Appendix B.

5.2. Human Evaluation

We conducted further investigation into the quality of synthetic multi-hop questions with the assistance of native English speakers[8]. We compared the questions generated by E2EQR and the strongest baseline model, BART, against ground-truth questions. We randomly selected 40 document sets for each complexity level, and the three different questions based on each document set were evaluated according to three criteria: *Fluency* assesses whether the question is logically constructed and adheres to correct grammar. Each question is assigned a score of 1 (poor), 3 (acceptable), or 5 (good). *Complexity* indicates the number of documents relevant to the question. The raters list the indices of any documents that are referenced to understand the question, and we report the average number of selected documents. *Answer Matching* determines whether the question asks about the input answer.

Table 2 presents the average scores rated by two evaluators. In terms of fluency, the questions generated by both multi-hop QG models are assessed as fluent and grammatically sound, akin to ground-truth questions. Notably, E2EQR demonstrates superior performance over BART in answer matching criteria in 3- and 4-hop QG, despite both models receiving similar complexity scores. The BART model is trained using end-to-end teacher forcing with ground-truth questions. Consequently, the model's capacity to discern relationships among input documents and generate questions pertinent to the input answer is constrained. Conversely, our model increases question complexity based on the documents and bridge entities entered at each rewriting step, thereby generating questions consistent with the input answer. Given that our model produces appropriate multi-hop QA pairs, it also proves effective in augmenting data for multi-hop QA in Section 6.1.

Fluency	Complexity Growth	Relevance
4.60	87.9%	86.8%

Table 3: Human evaluation of the question rewriting in our model.

We additionally verified whether E2EQR successfully rewrites the intermediate question generated in the preceding step, taking into account the newly

[8] We enlisted human evaluators from Amazon Mechanical Turk (https://www.mturk.com) and Upwork (https://www.upwork.com).

entered document. Among the final and intermediate questions generated by E2EQR, 100 pairs of questions, before and after rewriting, were randomly selected. The rewritten questions were then evaluated by two English native speakers based on three criteria: *Fluency*, *Complexity Growth*, and *Relevance*. Complexity growth assesses whether the rewritten question exhibits greater complexity than its predecessor. Relevance indicates whether the question rewriting process references the content of the input document.

As shown in Table 3, the intermediate questions generated by E2EQR display both logical and grammatical correctness. Additionally, over three-quarters of the questions were successfully rewritten based on the provided documents, resulting in greater complexity compared to their predecessors. Generating intermediate questions enables the model to logically enhance the complexity of the question and facilitate better understanding of the results. Furthermore, multi-hop questions feature intricate sentence structures, posing challenges for error detection. Therefore, leveraging intermediate questions enables identification and filtering of incorrect multi-hop questions at an early stage.

6. Analysis and Discussion

6.1. Data Augmentation for Question Answering

We assess the efficacy of synthetic data generated by E2EQR for training multi-hop QA models. First, we trained the models using the training$_{seen}$ set of MuSiQue. The models then generated questions from unseen documents in the training$_{unseen}$ set.

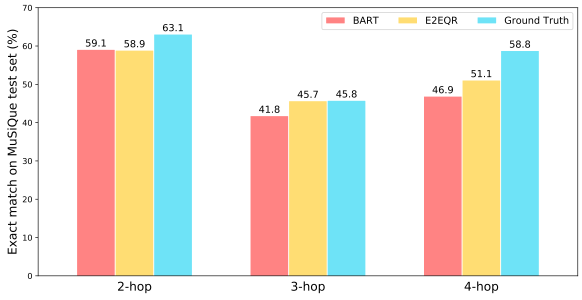


Figure 3: The performance of three QA models trained on the synthetic data generated by BART and E2EQR and the MuSiQue training$_{unseen}$ set (Ground Truth), respectively.

Figure 3 describes the best performance of QA models trained on the synthetic data of E2EQR and BART, and the ground truth, respectively. We implemented the QA models based on T5 (Raffel et al., 2020). According to the results, the QA model trained on our data achieves 3.9 and 4.2 higher performance in 3- and 4-hop examples, respectively, than the model trained on the data of BART. In the 2-hop examples, both QG models have the similar effects in training QA models.

6.2. Ablation Study

Model	2-hop	3-hop	4-hop
E2EQR	44.61	40.31	38.41
E2EQR w/o Accumulated SA	43.32	39.94	35.14
E2EQR w/o Accumulated CA	41.84	36.69	35.35

Table 4: The ablation study for E2EQR. The best ROUGE-L score is reported.

We further investigated the effects of the accumulated SA and CA mechanisms on the performance of E2EQR. Table 4 shows the performance of E2EQR trained without each component. For each case, we used the original SA or CA mechanism instead of the accumulated ones. Without the accumulated SA, a performance degradation of 1.29, 0.37 and 3.27 points is observed in 2-, 3- and 4-hop QG, respectively. In addition, the model without the accumulated CA mechanism has a more significant performance drop, especially in 2- and 3-hop QG. Consequently, we prove that E2EQR rewrites the questions relying on the representations of the previous input documents and intermediate questions, which are transferred through the accumulated CA and SA.

6.3. Analysis of Training Strategies

Training Strategy		2-hop	3-hop	4-hop
1. Standard Training		43.00	40.99	34.03
Curriculum Learning	2. Step-by-step	28.87	38.69	33.95
	3. Cumulative	43.67	38.99	36.66
	4. Adaptive *(ours)*	44.61	40.31	38.41

Table 5: Performance of E2EQR trained using different strategies.

We also experimented with various training strategies to enable E2EQR to effectively learn multi-hop questions with diverse complexities. Table 5 compares the ROUGE-L scores of E2EQR models trained by four different training strategies.

1. In standard training procedures, the model is trained uniformly across all examples, regardless of the question complexity. Contrasting our proposed approach, this method exhibits lower performance in generating 4-hop questions. This outcome indicates that simultaneous learning from examples of various com-

Example 1
[Document A] African-American candidates for President of the United States
In 1888 Frederick Douglass was invited to speak at the Republican National Convention. Afterward during the roll call vote, he received one vote, so was nominally a candidate for the presidency. In those years, the candidates for the presidency and vice presidency were chosen by state representatives voting at the nominating convention. Many decisions were made by negotiations of state and party leaders behind closed doors. Douglass was not a serious candidate in contemporary terms.
[Document B] Helen Pitts Douglass
Helen Pitts Douglass (1838–1903) was an American suffragist and abolitionist, known for being the second wife of Frederick Douglass. She also created the Frederick Douglass Memorial and Historical Association.
Answer: Helen Pitts Douglass
DP-Graph: who was the child of the person responsible for the political party with douglass reid ?
CQG: who was the spouse of the president of the united states as a member?
MulQG: who are the two leaders of the opposition in the province where douglass is located ?
BART: Who is the spouse of the person who was invited to speak at the Republican National Convention in 1888?
E2EQR: Who is the spouse of the person who was invited to speak at the 1888 Republican convention?
Ground Truth: Who is the spouse of the first nominated African American presidential candidate?
Example 2
[Document A] Rio Linda High School
Rio Linda High School is a high school located in Rio Linda, Sacramento, CA. It has an enrollment of 2,035 students. It is part of the Twin Rivers Unified School District, and was formerly part of the Grant Unified School District.
[Document B] Area code 951
Area code 951 is a California telephone area code that was split from area code 909 on July 17, 2004. It covers western Riverside County, including, Beaumont, Corona, Canyon Lake, Riverside, Temescal Canyon, ...
[Document C] History of Sacramento, California
The history of Sacramento, California, began with its founding by Samuel Brannan and John Augustus Sutter, Jr. in 1848 around an embarcadero that his father, John Sutter, Sr. constructed at the confluence of the American and Sacramento Rivers a few years prior.
[Document D] California Gold Rush
Rumors of the discovery of gold were confirmed in March 1848 by San Francisco newspaper publisher and merchant Samuel Brannan. Brannan hurriedly set up a store to sell gold prospecting supplies, ...
Answer: Rio Linda
DP-Graph: what city is the capital of the county where the town of linda is headquartered ?
CQG: what is the administrative territorial entity where the gold rush is located?
MulQG: what is the area code for the state where area code 951 is located ?
BART: What is the capital of the area code 951 of the state where the California Gold Rush began?
E2EQR: What city shares a border with the county where the person who started the Gold Rush in the US city having area code 951 worked in?
Ground Truth: What shares a border with the city where the person who went to the state where the 951 area code is used during the gold rush works?

Table 6: Examples of multi-hop questions generated based on two and four documents.

plexity levels results in inadequate comprehension of highly complex questions.

2. In step-by-step curriculum learning, the model is optimized sequentially, starting from generating simple questions and progressing to generating complex ones. We trained the model by configuring the hyperparameters of Algorithm 1 as follows: $\{\gamma_{low} = 0, \gamma_{high} = 0, \rho = 0\}$. As shown in the results, the model experiences catastrophic forgetting with respect to the 2-hop questions initially learned.

3. In cumulative curriculum learning, the model sequentially learns from simple to complex examples, akin to the previous method. However, in this approach, the easier examples on which the model has already been optimized are also integrated into subsequent training process. We trained the model with the setting $\{\gamma_{low} = 1, \gamma_{high} = 0, \rho = 0\}$. While this method facilitates effective learning of questions with diverse complexities, it is imperative to mitigate overfitting to the simpler questions.

4. In our adaptive curriculum learning, the model learns a small number of complex questions even during the initial iterations that primarily focus on easier examples. This approach ensures that the model trains on easier examples while also considering the necessity for rewriting to generate more complex questions. As a result, our method empowers the model to achieve robust and balanced performance across multiple complexity levels.

6.4. Case Study

Table 6 shows questions generated by different multi-hop QG models based on the same context documents. *Example 1* illustrates the case of generating a 2-hop question based on two given documents. The questions generated by BART and E2EQR suggest correct reasoning processes that ultimately lead to the input answer.

Example 2 depicts a more challenging scenario as the models are required to generate 4-hop questions that encompass the content of four distinct documents. BART generates questions by using key entities present in *Document B* and *Document D*. Given the question, *"the area code 951"* can be replaced with *"California"* based on information from *Document B*, while *"where the California Gold*

Rush began" can be substituted with *"San Francisco"*, as indicated in *Document D*. As a consequence of question simplification, the question generated by BART is equivalent to *"What is the capital of California of San Francisco?"*, which is neither logical nor consistent with the answer.

In contrast, the question generated by E2EQR facilitates sequential simplification based on the given documents. *"What city shares a border with the county where the person who started the Gold Rush in California worked in?"* transforms into *"What city shares a border with the county where Samuel Brannan worked in?"*, further simplifying to *"What city shares a border with Sacramento?"* It can be confirmed that the finally obtained 1-hop question matches the input answer.

Answer: July 22, 1864
[Document A] Battle of Atlanta The Battle of Atlanta was a battle of the Atlanta Campaign fought during the American Civil War on July 22, 1864, just southeast of Atlanta, Georgia. Continuing their summer campaign to seize the important rail and supply center of Atlanta, Union forces commanded by William Tecumseh Sherman overwhelmed and defeated Confederate forces defending the city under John Bell Hood. . . . **Intermediate Question (1-hop)**: When did the battle of Atlanta happen?
[Document B] List of municipalities in Georgia The largest municipality by population in Georgia is Atlanta with 420,003 residents, and the smallest municipality by population is Edge Hill with 24 residents. . . . **Intermediate Question (2-hop)**: When did the Battle of the largest municipality in Georgia happen?
[Document C] WEKL WEKL, known on-air as "K-Love", is a Contemporary Christian radio station in the United States, licensed by the Federal Communications Commission (FCC) to Augusta, Georgia, broadcasting on 102.3 MHz with an ERP of 1.5 kW. . . . **Final Question (3-hop)**: When did the Battle of the largest municipality in the state WEKL broadcasts in happen?

Table 7: An example of question rewriting in E2EQR. The expressions referring to the same object in the same color.

Table 7 presents an example of question rewriting using E2EQR. The 1-hop question was generated based on *Document A* and contains the bridge entity *"Atlanta"* that connects *Document A* and *B*. Next, our model rewrote the question to a 2-hop question using the additional information about *"Atlanta"* in *Document B* and included the bridge entity *"Georgia"* connecting *Document B* and *C*. In the final step, *"Georgia"* in the previous question was replaced with the information about *"WEKL"* appeared in *Document C*.

7. Conclusion

This paper introduces a novel multi-hop question generation model that sequentially enhances question complexity based on the input documents. Employing a step-by-step question rewriting strategy, the proposed model generates multi-hop questions, yet remains trainable end-to-end without reliance on labeled intermediate questions. Experimental results confirm the model's effectiveness in generating complex questions requiring inference across multiple documents, while accurately aligning with input answers. Furthermore, the synthetic QA pairs generated by our model are also beneficial when used as training data for multi-hop QA models.

8. Ethics Statement

This study utilized commonly used QA datasets devoid of ethical concerns. Furthermore, we ensured ethical practices by providing fair compensation to human evaluators recruited from Amazon Mechanical Turk and Upwork.

9. Acknowledgements

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2022-0-00653, Voice Phishing Information Collection and Processing and Development of a Big Data Investigation Support System) This work was also supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2019-0-01906, Artificial Intelligence Graduate School Program(POSTECH)).

10. Bibliographical References

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic qa corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Ying-Hong Chan and Yao-Chung Fan. 2019. A recurrent bert-based model for question generation. In *Proceedings of the 2nd workshop on machine reading for question answering*, pages 154–162.

Yi Cheng, Siyao Li, Bang Liu, Ruihui Zhao, Sujian Li, Chenghua Lin, and Yefeng Zheng. 2021. Guiding the growth: Difficulty-controllable question generation through step-by-step rewriting. 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 5968–5978.

Xinya Du, Junru Shao, and Claire Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352.

Zichu Fei, Qi Zhang, Tao Gui, Di Liang, Sirui Wang, Wei Wu, and Xuan-Jing Huang. 2022. Cqg: A simple and effective controlled generation framework for multi-hop question generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6896–6906.

Michael Heilman. 2011. *Automatic factual question generation from text*. Ph.D. thesis, Carnegie Mellon University.

Michael Heilman and Noah A Smith. 2010. Good question! statistical ranking for question generation. In *Human language technologies: The 2010 annual conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Karen Mazidi and Rodney Nielsen. 2014. Linguistic considerations in automatic question generation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 321–326.

Liangming Pan, Wenhu Chen, Wenhan Xiong, Min-Yen Kan, and William Yang Wang. 2021. Unsupervised multi-hop question answering by question generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5866–5880.

Liangming Pan, Yuxi Xie, Yansong Feng, Tat-Seng Chua, and Min-Yen Kan. 2020. Semantic graphs for generating deep questions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1463–1475.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

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*, pages 2383–2392.

Dan Su, Yan Xu, Wenliang Dai, Ziwei Ji, Tiezheng Yu, and Pascale Fung. 2020. Multi-hop question generation with graph convolutional network. *Proceedings of Findings of the Association for Computational Linguistics (EMNLP)*.

Xingwu Sun, Jing Liu, Yajuan Lyu, Wei He, Yanjun Ma, and Shi Wang. 2018. Answer-focused and position-aware neural question generation. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3930–3939.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2016. Newsqa: A machine comprehension dataset. *arXiv preprint arXiv:1611.09830*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Liuyin Wang, Zihan Xu, Zibo Lin, Haitao Zheng, and Ying Shen. 2020. Answer-driven deep question generation based on reinforcement learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5159–5170.

John H Wolfe. 1976. Automatic question generation from text-an aid to independent study. In *Proceedings of the ACM SIGCSE-SIGCUE technical symposium on Computer science and education*, pages 104–112.

Yuxi Xie, Liangming Pan, Dongzhe Wang, Min-Yen Kan, and Yansong Feng. 2020. Exploring question-specific rewards for generating deep questions. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2534–2546.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 3901–3910.

11. Language Resource References

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*, 10:539–554.

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

A. LLM-based Multi-hop Question Generation

Recently, large language models (LLMs) have demonstrated remarkable performance across various NLP tasks. We also investigated whether LLMs are capable of generating multi-hop questions through in-context learning. In this experiment, we assign two tasks to GPT-3.5[9]: standard multi-hop QG, where the model generates multi-hop questions based on the given documents and answers, and incremental multi-hop QG, wherein the model generates questions sequentially from 1-hop to the target complexity level. In the incremental generation, we require the model to specify the index of the referenced document to generate or revise the question. The prompt templates are described in Table 8 and 9.

Generate a multi-hop question for the given answer which requires reference to all of the given documents.
(Example1)
Document1: {{document1}}
Document2: {{document2}}
Document3: {{document3}}
Answer: {{answer}}
Question: {{question}}

...

(Example4)
Document1: {{document1}}
Document2: {{document2}}
Document3: {{document3}}
Answer: {{answer}}
Question:

Table 8: Prompt template used for standard generation. We attached three exemplars randomly selected from the MuSiQue training set.

Generate a multi-hop question for the given answer which requires reference to all of the given documents.
(Example1)
Document1: {{document1}}
Document2: {{document2}}
Document3: {{document3}}
Answer: {{answer}}
1-hop question (using Document$i$): {{1-hop question}}
2-hop question (using Document$j$): {{2-hop question}}
3-hop question (using Document$k$): {{3-hop question}}
...

(Example4)
Document1: {{document1}}
Document2: {{document2}}
Document3: {{document3}}
Answer: {{answer}}

Table 9: Prompt template used for incremental generation. We randomly selected three exemplars from the MuSiQue training set and manually labeled their intermediate questions.

According to the results presented in Table 10, GPT-3.5 exhibits inferior performance compared to E2EQR. However, GPT-3.5 demonstrates better performance in incremental generation, wherein the model progressively enhances question complexity, than in standard generation. This observation indicates the effectiveness of sequential question rewriting approach to generate complex questions, even for advanced LLMs.

We analyzed the prevalent error cases in the

[9] https://platform.openai.com/

Model		2-hop	3-hop	4-hop
GPT-3.5	Standard	23.54	22.92	20.64
	Incremental	26.04	25.98	21.86
E2EQR		44.01	40.04	36.98

Table 10: ROUGE-L scores of GPT-3.5 models.

Answer: *Tamaulipas*
[Document 1] The Municipality of Nuevo Laredo is located in the Mexican state of *Tamaulipas*. ... **[Document 2]** The Ciudad Deportiva (Šports City)̈ is a sports complex in Nuevo Laredo, Mexico. It is home to the Tecolotes de Nuevo Laredo Mexican Baseball League team and ...
Standard Generation
Which Mexican state is home to the municipality containing Nuevo Laredo and Ciudad Deportiva?
Incremental Generation
1-hop question (using Document1): In which Mexican state is Nuevo Laredo located? 2-hop question (using Document2): In which Mexican state can one find the Ciudad Deportiva, home to the Tecolotes de Nuevo Laredo?
Ground Truth
What administrative territorial entity is the owner of Ciudad Deportiva located?
Answer: *Milledgeville*
[Document 1] Blackberry Smoke is an American rock band from Atlanta, Georgia, United States. ... **[Document 2]** WEKL, known on-air as "102.3 K-Love", is a Contemporary Christian radio station in the United States, ... **[Document 3]** Georgia has had five different capitals in its history. The first was Savannah, the seat of government during British colonial rule, followed by Augusta, Louisville, *Milledgeville*, and Atlanta, the capital city from 1868 to the present day. ...
Standard Generation
Where is the location of the studios for the radio station in Augusta where the American rock band Blackberry Smoke originates from?
Incremental Generation
1-hop question (using Document3): What city served as the capital of Georgia before Atlanta? 2-hop question (using Document2): What city in Georgia is the 102.3 K-Love radio station located in? 3-hop question (using Document1): What city in Georgia is the band Blackberry Smoke from?
Ground Truth
What was the capital of the state where WEKL operates, before the city where Blackberry Smoke was formed?

Table 11: Examples of 2- and 3-hop questions generated by GPT-3.5.

generation results of GPT-3.5. Table 11 illustrates representative examples of 2- and 3-hop generated questions. In the standard generation, the model can be seen to successfully generates questions that incorporate content derived from multiple documents. However, the model tends to produce questions that can be resolved by referring to a single document. For example, the 2-hop question generated in standard setting includes *"Ciuda Deportiva"* from *Document 2*, yet can be answered by reading only *Document 1*. Similar outcomes are observed in 3-hop QG, where the question does not even match the input answer. We believe that while GPT-3.5 recognizes and mimic patterns from few-shot examples, it encounters limitations in logically generating multi-hop questions that correspond appropriately with the input answers.

Furthermore, during incremental generation, GPT-3.5 encounters difficulties in generating 3- and 4-hop questions, frequently formulating the questions that primarily focus on the content of the last referenced document. As shown in the 3-hop generation, the model inserts information from a newly referenced document into the question without retaining the main content of the previous question. In conclusion, in-context learning is not sufficient to achieve the objective of generating logically structured multi-hop questions that match the input answers, and our model is superior at multi-hop QG tasks although it requires supervised learning.

B. Automatic Evaluation on HotpotQA

Model	BLEU-4	METEOR	ROUGE-L
DP-Graph (Pan et al., 2020)	15.53	20.15	36.94
MulQG (Su et al., 2020)	15.20	20.51	35.30
CQG (Fei et al., 2022)	21.46	24.97	39.61
BART (Lewis et al., 2020)	21.77	29.40	43.01
E2EQR	21.73	28.47	41.34

Table 12: Automatic evaluation results on HotpotQA.

We compared the performance of various multi-hop QG models on HotpotQA, which contains two question types: *bridge* and *comparison*. Because our model cannot generate the comparison question using question rewriting, we handled these as 1-hop questions. As presented in Table 12, the proposed model achieves slightly lower performance than BART but outperforms the remaining baselines. It seems that BART performs better than E2EQR because the supporting facts in the HotpotQA dataset are shorter than those from the MuSiQue dataset, and two-hop QG is sufficiently possible with a seq2seq model.