Prompting Few-shot Multi-hop Question Generation via Comprehending Type-aware Semantics

Zefeng Lin, Weidong Chen, Yan Song†, Yongdong Zhang
University of Science and Technology of China
zflin@mail.ustc.edu.cn chenweidong@ustc.edu.cn
clksong@gmail.com zhyd73@ustc.edu.cn

Abstract

Given several documents, multi-hop question generation (MQG) is a task aims to generate complicated questions that require reasoning over multiple pieces of these documents to find the answer. To perform this task, existing studies focus on designing advanced architectures to locate essential keywords or sentences in multiple documents and then generate questions accordingly, where they normally do not note that question types could provide crucial hints for extracting key information from the documents for MQG. In general, supervised approaches are used that rely on large annotated data, which is not available in many low-resource scenarios and thus makes MQG hard in these domains. Consider the recent success of large language models (LLMs) on natural language processing tasks using limited labeled data under few-shot settings, in this paper, we propose an approach named type-aware semantics extraction-based chain-of-thought method (TASE-CoT) for few-shot MQG. Specifically, our approach firstly extracts question types and essential semantic phrases from the given documents and the answer. Then, we design a three-step CoT template to leverage the extracted question type and semantic phrases to predict multi-hop questions. Extensive experiments and the results demonstrate the effectiveness of our approach and the proposed modules.[1]

1 Introduction

Question generation (QG) aims to generate questions that are relevant to the given document. It is a vital task in the field of question answering (QA) owing to its wide applications, e.g., helping chatbots start conversations with intriguing questions (Skjuve et al., 2022; Janssen et al., 2022). Most existing approaches for QG (Du et al., 2017; Zhou et al., 2018; Kim et al., 2019; Fei et al., 2021; Mulla and Gharpure, 2023) focus on generating simple one-hop questions based on a single document, which cannot cover the cases that need complicated multi-hop questions requiring a deep understanding of multiple documents to answer. Under this circumstance, multi-hop question generation (MQG), which aims to generate multi-hop questions where answering the questions requires reasoning over multiple documents, has attracted increasing interest from both the academia and industry community (Pan et al., 2020; Sachan et al., 2020; Ji et al., 2021; Su et al., 2022a; Fei et al., 2022; Yu et al., 2023; Xia et al., 2023).

Existing studies (Pan et al., 2020; Fei et al., 2022; Xia et al., 2023) on MQG usually regard semantic phrases as nodes and build graphs over them. They utilize graph neural networks (GNN) or node clas-

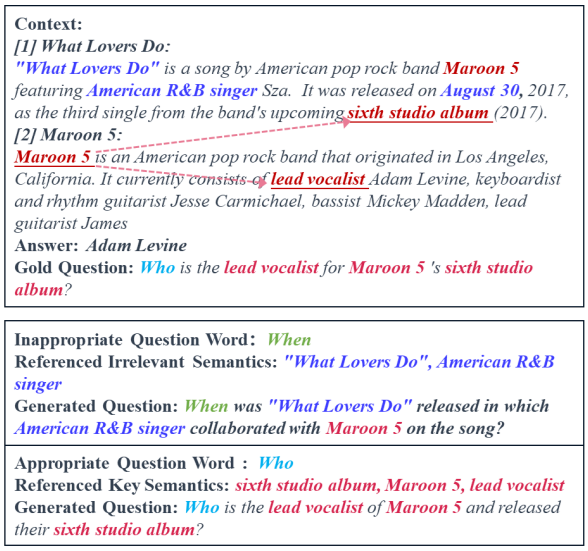

Figure 1: The figure presents examples where the question type affects the process of generating questions. In this example, the first model selects the inappropriate question type "*when*", and thus extracts irrelevant semantics. On the contrary, the second model selects the appropriate question type "*who*", and thus extracts the appropriate key semantics to generate questions. Important semantic phrases are highlighted in red color.

†Corresponding author.

[1]The source code and relevant resources of the paper are available at https://github.com/synlp/TASE-CoT.

Findings of the Association for Computational Linguistics: NAACL 2024, pages 3730–3740
June 16-21, 2024 ©2024 Association for Computational Linguistics

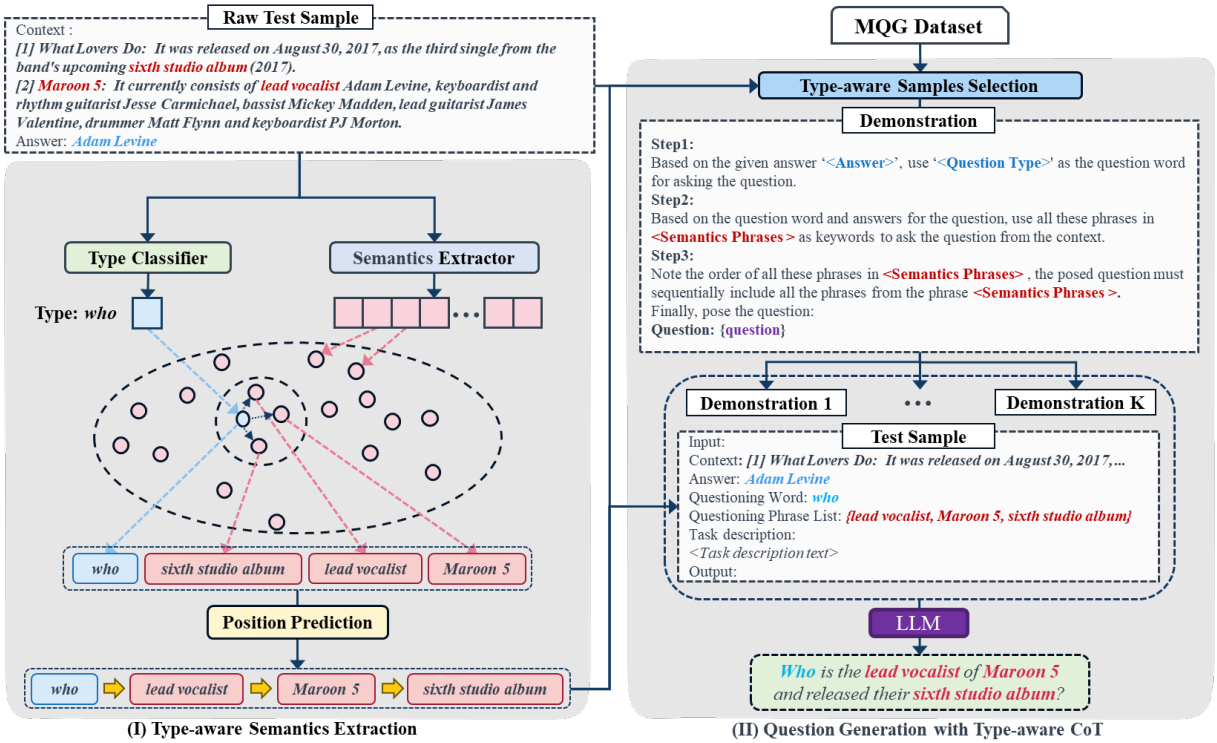

Figure 2: The figure shows the overall pipeline of the TASE-CoT approach. It consists of two steps, namely, type-aware semantics extraction and question generation with type-aware CoT, which are presented on the left- and right-hand sides of the figure, respectively. Example input and prompt templates are presented for better illustration.

sification to extract reasoning chains or keywords from multiple documents. Then, they use the reasoning chains and keywords to help MQG models generate relevant multi-hop questions. These approaches generally require a large amount of labeled training data to learn a well-performing MQG model and are hard to apply to low-resource situations. Owing to the recent success of large language models (LLMs) on few-shot learning (Brown et al., 2020; Wei et al., 2022), it is intuitive to perform MQG with LLMs under the few-shot setting when the training resources are limited. However, prompting LLMs with straightforward instructions to generate multi-hop questions is not trivial to obtain satisfying performance. LLMs are inefficient at extracting and utilizing key semantic phrases that are essential to produce a multi-hop question. Consider prompting LLMs with relevant information is demonstrated to be effective for many NLP tasks (Wei et al., 2022; Su et al., 2022b; Liang et al., 2023; Fei et al., 2023), a carefully designed prompting strategy with semantic information is expected to be helpful for better MQG. Meanwhile, we notice that the question type provides hints in extracting key semantic phrases and producing multi-hop questions that satisfy human preferences, as illustrated in the examples in Figure 1.

In this paper, we propose an approach named type-aware semantics extraction-based chain-of-thought (TASE-CoT) for MQG. Our approach utilizes a type-aware semantics extraction (TASE) model to extract question types and key semantic phrases, which are utilized in the type-aware chain-of-thought (CoT) framework to generate questions. Specifically, TASE firstly predicts the question type based on the given answer and context documents. Then, it utilizes the question type to extract the key semantic phrases. Type-aware CoT constructs a CoT prompt based on the obtained question type and semantic phrases. The CoT prompt breaks the MQG process into multiple steps according to the general process when humans produce questions. The proposed module selects training set samples with similar question types to construct few-shot demonstration examples, which further enhances LLMs' understanding of the CoT prompt and thus improves model performance. Extensive experiments illustrate the effectiveness of our approach and each proposed module, and it achieves state-of-the-art performance on few-shot MQG and comparable performance to fine-tuning methods.

2 The Approach

The overall architecture of our approach is illustrated in Figure 2. It generates the question $\hat{q}$

with the given set of N context documents $C = \{d_1, ..., d_N\}$ and an answer a related to C, where a is the answer to the generated question $\widehat{q}$ using information from at least two documents in C. Our approach consists of two parts, the first TASE module f_1 (see the left of Figure 2) extracts the question type $\widehat{t}$ and semantic phrases $\mathcal{S}$ that provide essential information for MQG. The second part f_2 leverages the extracted t and $\mathcal{S}$ to construct the type-aware CoT and use it to instruct LLMs to generate multi-hop questions. Thus, the overall objective of our proposed framework is defined as follows:

$$\widehat{q} = f_2(f_1(C, a), \mathcal{D}, C, a) \quad (1)$$

where $\mathcal{D}$ is the training set that is used to extract demonstration examples to facilitate few-shot learning. The details of the two steps are illustrated in the following texts.

2.1 Type-aware Semantics Extraction

For MQG, existing studies (Fei et al., 2022; Xia et al., 2023) demonstrate that the semantic phrases that are relevant to the given answer contribute to generating high-quality questions. Meanwhile, the question type provides important hints on locating these relevant and important semantic phrases. Therefore, we propose the type-aware semantics extraction method. This method first predicts the question type and then uses the question type to locate important semantic phrases. Finally, the semantic phrases are sorted to guide the model to generate questions in a specific order. The details of the question type classifier, the semantic phrase extractor, and the semantic phrase ordering process are illustrated in the following text.

Question Type Classifier We use the encoder of T5 (Raffel et al., 2020) as our question type classier. It takes the answer a and the context documents $d_1 \cdots d_N$ as the input and predicts the type $\widehat{t}$. Specifically, we concatenate a and $d_1 \cdots d_N$ and feed the resulting text $[a; d_1 \cdots d_N]$ as the input to the classifier. The T5 encoder $\texttt{Encoder}_{TC}$ computes the hidden vectors for the input, and we apply a $\texttt{MeanPooling}$ operation to the hidden vectors to obtain the question type representation $\mathbf{h}_t$. The process is formulated as

$$\mathbf{h}_t = \texttt{MeanPooling}(\texttt{Encoder}_{TC}(a; d_1 \cdots d_N)) \quad (2)$$

Afterwards, we employ a linear projection layer with $\texttt{Softmax}$ function to $\mathbf{h}_t$ and predict the question types $\hat{t}$:

$$\widehat{t} = \texttt{Softmax}(\texttt{Linear}(\mathbf{h}_t)) \quad (3)$$

where $\widehat{t}$ is used in the subsequent process to identify important semantic phrases for MQG.

To train the question type classifier, we define the question types and collect the training data through the following process. Motivated by the observation that the first word in English questions generally determines the content they are asking, we collect the first word of all questions in the MQG training set as the raw type set. Then, we manually go through the set and filter out the types that do not make sense. The resulting question type set contains 22 question types and an additional "other" type, which the 22 general types are elaborated in Table 1. Finally, we extract the gold standard question type corresponding to the first word in the gold standard question in the MQG training set and use it to train the question type classifier.

	Question Types
wh-	*how, what, when, where, which, who, whom, whose*
be	*are, is, was*
do	*did, do, does*
have	*had, have, has*
will/can	*can, could, should, will, would*

Table 1: The table shows 22 general question types used in our approach. The question types are grouped into five categories for better illustration; the five categories are not used in our approach.

Semantic Phrase Extractor The semantics extractor aims to locate important semantic phrases that contribute to MQG based on the predicted question type $\widehat{t}$. Following Xia et al. (2023), we regard the important semantics phrase extraction as a sequence labeling task, where each semantic phrase is annotated by a binary label indicating whether it is an important semantics phrase. We use a Transformer-based approach as the semantics extractor. The Transformer encoder $\texttt{Encoder}_{SE}$ takes the concatenation of the answer a and the documents $d_1 \cdots d_N$, and computes the hidden vector for each semantic phrase. The l-th hidden vector for the l-th semantic phrase is denoted as $\mathbf{u}_l$. The process is formulated as

$$\mathbf{u}_1 \cdots \mathbf{u}_L = \texttt{Encoder}_{SE}(a; d_1 \cdots d_N) \quad (4)$$

where L is the total number of semantic phrases in the documents. Next, for each $\mathbf{u}_l$, we add it to the question type representation $\mathbf{h}_t$ and feed the resulting vector $\mathbf{o}_l = \mathbf{h}_t + \mathbf{u}_l$ into a fully connected projection layer with `Softmax` classifier. Thus, the important phrase label $\widehat{z}_l$ is obtained by

$$\widehat{z}_l = \texttt{Softmax}(\texttt{Linear}(\mathbf{o}_l)) \quad (5)$$

We compute $\widehat{z}_1 \cdots \widehat{z}_L$ for all semantic phrases and extract the important ones accordingly. We denote these semantic phrases as $s_1 \cdots s_M$, where M is their total number and the representation of the m-th semantics phrase s_m is $\mathbf{o}_m$.

To train the semantics extractor, it requires gold standard important semantic phrases. We regard the ones shared by the gold standard question and the documents as the gold standard important phrases.

Semantic Phrase Ordering We observe that the order of the extracted important semantic phrases provides essential hints for generating high-quality multi-hop questions. This motivates us to perform semantics ordering to find the appropriate order of the semantic phrase. We refer to the approach proposed by Li et al. (2022) to predict the order of semantic phrases, whose effectiveness is demonstrated in leveraging the order of different semantic phrases to improve text generation. Overall, our approach contains two steps. The first step computes the position representation of each semantic phrase using an attention mechanism; the second step uses a Transformer decoder to generate the original semantic phrases one by one. The order of the generated semantic phrases indicates their satisfactory order in the multi-hop questions.

Specifically, in the first step, we use the standard positional embedding matrix $\mathbf{E}_{POS}$ for Transformer (Vaswani et al., 2017) and use it as the keys and values in the attention mechanism, where the representation $\mathbf{o}_m$ of the semantics phrase s_m is used as the query. Therefore, the position representation $\mathbf{p}_m$ of s_m is computed by

$$\mathbf{p}_m = \texttt{Softmax}(\mathbf{o}_m \cdot \mathbf{E}_{POS}^{\top}) \cdot \mathbf{E}_{POS} \quad (6)$$

where $\mathbf{E}_{POS}^{\top}$ means the transpose of $\mathbf{E}_{POS}$. Afterwards, we add $\mathbf{p}_m$ to $\mathbf{o}_m$ and feed the resulting vector into the Transformer decoder in the second step. The decoder predicts the semantic phrase $s'_1 \cdots s'_M$ following the standard process, where $s'_1 \cdots s'_M$ are the reordering of $s_1 \cdots s_M$ and its order is used to help the following MQG. To train the semantics ordering model, we use the order of the semantic phrase in the gold standard question as the gold standard and optimize the model accordingly.

2.2 Question Generation with Type-aware CoT

Existing studies have shown that the quality of demonstration examples is essential for achieving good performance under few-shot settings (Zhang et al., 2022). To obtain high-quality demonstration examples, we propose a few-shot CoT prompt construction approach that consists of two steps. The first is type-aware sample selection and the second is question generation with CoT. The details of the two steps are illustrated as follows.

Type-aware Sample Selection The goal of sample selection is to extract demonstration examples that are similar to the test instance, so that the LLM is able to learn relevant information to process the test instances from the given examples. Intuitively, the more similar the demonstration examples are to the test instances, the better the examples are. Given the question type is an essential feature that could help generate high-quality question, we propose to select demonstration examples that share the same question type with the test instance. Thus, for a test instance, we select demonstration examples using the following process.

Consider the quality of text representation plays an essential role in text understanding (Conneau et al., 2017; Song et al., 2017; Song and Shi, 2018; Han et al., 2018; Sileo et al., 2019; Song et al., 2021; Gan et al., 2023), we first use the Sentence-BERT model (Reimers and Gurevych, 2019), which is demonstrated to be effective in extracting sentence-level representations, to encode the test instance and all training instances. Sentence-BERT encodes the combination of the answer a, the question type $\widehat{t}$, and the semantic phrase $s'_1 \cdots s'_M$ of the test instance and obtain its representation $\mathbf{x}$. We perform the same process to compute the representation of the i-th training instances $\mathbf{x}_i$ whose question type is identical to $\widehat{t}$, where their semantic phrases are obtained from the type-aware semantics extraction process illustrated in Section 2.1. For each training instance, we compute the cosine similarity between $\mathbf{x}$ and $\mathbf{x}_i$ and select K training instances with the top K highest similarity scores as the demonstration examples.

Question Generation with CoT To perform CoT, it is required to have a step-by-step process

Dataset Name	Train	Dev	Test
HotpotQA	89,947	500	7,405
2WikiMultiHopQA	167,454	12,576	12,576

Table 2: The table shows the number of instances in the two benchmark datasets for MQG.

to generate the question for the demonstration examples. Given that it is expensive to ask human annotators to annotate the process, we propose using a CoT template that presents the general process of generating questions with the given answers and documents. Generally, when humans propose a multi-hop question based on multiple documents and predefined answers, they first determine the questioning type based on the answer, then select appropriate key semantic phrases from multiple documents, and finally formulate the question with the semantic phrases appearing in a particular order. Motivated by the process of producing questions by humans, we design a three-step template that leverages the question type and the semantic phrases. The template is illustrated on the right of Figure 2. The first step analyzes the question type with the given answer. The second step extracts important semantic phrases. The third step reorders the semantic phrase and instructs the LLM to predict the multi-hop question. We use the answer, the question type, and the semantic phrase in the demonstration examples to fill in the template and use them to instruct LLM to generate the question $\widehat{q}$ for the test instance.

3 Experiment Settings

3.1 Datasets

Following existing studies (Pan et al., 2020; Fei et al., 2022; Xia et al., 2023), we run experiments on two widely used English benchmark datasets named HotpotQA (Yang et al., 2018) and 2WikiMultiHopQA (Ho et al., 2020). We follow existing studies (Fei et al., 2022; Ho et al., 2020) to split the datasets into train/dev/test sets. We report the number of instances in the datasets in Table 2.

3.2 Baselines

As there are limited studies for few-shot MQG with LLMs, we adopt the following general prompting approaches in the few-shot setting as our baselines.

Vanilla Prompt (Brown et al., 2020) is the standard prompting method of in-context learning. In our implementation, we randomly select K examples from the training set to construct the demonstration examples for few-shot settings.

Random-CoT (Wei et al., 2022) is a naïve baseline where the K demonstration examples are randomly selected from the training set. We follow the design criteria of CoT in their study to construct the task description and demonstration.

Manual-CoT (Wei et al., 2022) is a CoT approach where the K demonstration examples are manually created. We construct the prompt for Manual-CoT based on the CoT template used in our approach. We try different variants with minor modifications of our CoT template and use the one with the best performance in experiments.

Auto-CoT (Zhang et al., 2022) is an approach that automatically generates CoT of demonstration examples. We apply this approach to the MQG task through the following process. We first encode all training instances using Sentence-BERT and obtain their representations. Then, we perform the clustering approach in Auto-CoT and choose the examples of different cluster centers to generate the reasoning chains in the demonstration examples.

Least-to-Most (Zhou et al., 2023) is an approach that prompts LLMs to solve problems step by step from easy to difficult. We apply this approach to the MQG task through the following process. We first decompose the MQG task into three steps corresponding to our CoT template, and then sequentially prompt LLMs to complete these steps, whereby the generation of the previous step is used to facilitate the generation of the current step.

CoT-SC (Wang et al., 2023) is a CoT approach that samples diverse reasoning paths generated by LLMs and chooses the most consistent answer by marginalizing these paths. We use the same prompt of Manual-CoT in CoT-SC.

3.3 Implementation Details

In Type-aware semantics extraction, we utilize the T5-base[2] (Raffel et al., 2020) as the encoder of the question type classifier and semantic extractor in our approach. We use the decoder of T5-base as the model for semantics ordering. For the Sentence-BERT used in our approach and baselines, we utilize the all-MiniLM-L6-v2[3] model. We

[2] https://huggingface.co/t5-base

[3] https://huggingface.co/Sentence-BERT/all-MiniLM-L6-v2

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
Vanilla Prompt (fixed)	29.27	16.73	11.33	7.92	13.94	22.03
Vanilla Prompt (variable)	22.48	14.50	10.86	8.10	11.68	21.06
Random-CoT (fixed)	33.99	20.97	14.15	10.21	16.78	24.62
Random-CoT (variable)	33.28	21.31	15.62	12.06	15.80	22.30
Manual-CoT	36.28	23.97	17.61	13.82	16.88	28.17
Least-to-Most	<u>39.33</u>	<u>27.23</u>	<u>20.06</u>	<u>15.06</u>	<u>20.96</u>	<u>32.51</u>
CoT-SC	35.69	24.09	18.51	14.88	18.36	30.31
Auto-CoT	27.96	19.58	14.68	11.35	14.99	31.65
TASE-CoT	**45.89**	**34.06**	**27.11**	**22.37**	**23.39**	**39.68**

Table 3: The table shows the experimental results of different models on HotpotQA with the few-shot setting. The best and second-best results are boldfaced and underlined, respectively.

utilize `gpt-3.5-turbo-1106`[4] from OpenAI API as the LLM to generate questions.

In the experiment, we set the number of demonstrations $K = 3$, where most methods achieve their best performance. In addition, we try two settings to select the demonstration examples for Vanilla Prompt and Random-CoT baselines. The first "fixed" configuration sets all K examples to be fixed for all test instances, while the second "variable" configuration sets them to be different for every test instance.

For evaluation metrics, we follow previous studies to employ the commonly used BLEU (Papineni et al., 2002), ROUGE-L (Lin, 2004), and METEOR (Lavie and Agarwal, 2007) as our automated evaluation metrics. Herein, BLEU and ROUGE-L are considered as precision and recall of n-gram matching to evaluate text generation tasks, respectively. METEOR is a comprehensive metric beyond exact matches, accounting for partial matches and variations in word order.

4 Results and Analysis

4.1 Main Results

We run our approach and baselines with the "fixed" and "variable" settings on the benchmark dataset. Table 3 shows the experimental results of different models. There are the following observations.

First, compared with baseline methods, our approach achieves better performance on HotpotQA datasets, which indicates the effectiveness of our approach. Second, we observe that all CoT-based prompting approaches outperform the Vanilla Prompting approach. This indicates the effectiveness of dividing the entire question generation process into subtasks in a CoT-style prompt, which is coherent with the conclusion shown in the previous work. Third, compared with Auto-CoT, Manual-CoT, which utilizes the question type information, achieves better performance, which shows the effectiveness of type-aware CoT. Fourth, comparing settings with the "fixed" or "variable" demonstration examples, we find that overall, the performance under the two settings is similar, which presents the robustness of our approach.

We further compare our approach with existing studies. The results on HotpotQA and 2WikiMultiHopQA are shown in Table 4 and 5, respectively. Herein, all existing studies on HotpotQA utilize supervised approaches, which are trained on the entire training data. We find that, with three demonstrations, our approach outperforms the majority of pre-trained models on the HotpotQA dataset. In addition, in the cross-domain setting of Table 5, our method outperforms all methods. Since our method first proposed the few-shot setting in the MQG task, to our best knowledge, there are no other few-shot MQG methods compared with our method. Therefore, we can view the TASE-CoT as a baseline for future work on the MQG task in the low-resource scenario.

4.2 Human Evaluation

We conducted the human evaluation by randomly sampling 300 examples from the test set of the HotpotQA dataset. Three annotators were asked to rate the questions generated by the prompting methods and the gold questions. The scale score is 1 to 5, where 1 denotes poor, and 5 denotes perfect. the rating mainly considers three aspects of fluency, relevance, and complexity, and follows the criteria

[4] https://platform.openai.com/docs/models/gpt-3-5

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
Full Training						
CGC-QG (Liu et al., 2019)	31.18	-	-	14.36	25.20	40.94
UniLM (Dong et al., 2019)	42.37	29.95	22.61	17.61	25.48	40.34
MuLQG (Su et al., 2020)	40.15	26.71	19.73	15.20	20.51	35.30
BART (Lewis et al., 2020)	41.41	30.90	24.39	19.75	25.20	36.13
SG-DQG (Pan et al., 2020)	40.55	27.21	20.13	15.53	20.15	36.94
IGND (Fei et al., 2021)	41.22	24.71	18.99	16.36	24.19	38.34
CQG (Fei et al., 2022)	49.71	37.04	29.93	25.09	27.45	41.83
MultiFactor (Xia et al., 2023)	**54.17**	**41.50**	**33.74**	**28.22**	**28.60**	**44.17**
Few-shot Evaluation						
TASE-CoT	45.89	34.06	27.11	22.37	23.39	39.68

Table 4: The table shows the comparison between the TASE-CoT approach and full-trained models on HotpotQA.

Method	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L
CQG	39.08	27.15	20.85	16.43	19.14	36.50
Auto-CoT	35.22	23.25	15.66	10.30	21.26	34.03
Least-to-Most	31.95	21.58	15.29	10.56	18.96	34.81
CoT-SC	26.97	16.39	11.12	8.02	16.36	22.79
TASE-CoT	45.38	31.35	23.15	17.65	27.00	37.42

Table 5: The table shows the comparison among different models on 2WikiMultiHopQA. The CQG model is trained on HotpotQA and tested on 2WikiMultiHopQA. Other few-shot methods are tested directly on 2WikiMultiHopQA.

Method	Fluence	Relevance	Complexity
Vanilla Prompt	3.19	2.74	2.26
Random-CoT	3.88	3.52	3.24
Manual-CoT	3.71	3.57	3.45
Auto-CoT	3.62	3.79	3.42
TASE-CoT	**4.20**	**4.17**	**4.10**
Ground Truth	4.93	4.89	4.95

Table 6: The table shows the human evaluation for different prompting methods on HotpotQA.

Method	BLEU-4	METEOR	ROUGE-L
TASE-CoT (Ours)	22.37	23.39	39.68
TASE-CoT (template 2)	20.49	24.25	38.84
TASE-CoT (template 3)	20.59	23.03	40.78
(a) w/o CoT	18.02	22.22	37.19
(b) type-aware→random	16.63	21.48	34.76
(c) w/o question type	15.76	20.87	37.17
(d) w/o semantics	15.39	18.12	33.12

Table 7: The table presents the experiment results of ablation study of our approach on HotpotQA, where different components are ablated.

of Fei et al. (2022). The score of each question is averaged over all annotators. We reported the results in Table 6, our approach outperforms all main baseline methods and obtains scores that are closer to the ground truth than other baselines.

4.3 Ablation Study

We conducted ablation studies to assess the effectiveness of components of our framework and reported the results in Table 7. The following are some observations. First, we exclude the CoT reasoning chain to test the necessity of CoT prompting. We observe a performance drop in the evaluated metrics, particularly a drop of 4.35 points in BLEU-4. This indicates that human question approach-based CoT prompting plays an important role in our framework. Second, we remove the type-aware demonstration selection method and randomly select training samples as the demonstration. The large decrease in the results indicates that our type-aware selection method can ensure our demonstrations have higher quality. We further remove the question type and semantics in the demonstrations respectively. The decreasing performance indicates that the question type and semantics information significantly affect the few-shot MQG. For analysis of sensitivity to templates, we also conducted extra experiments on different templates. The results

QT (Acc)	SPE (Acc)	QG (BLEU-4)
59.15	81.28	18.92
60.25	83.67	20.36
62.50	85.94	22.37

Table 8: The table shows the effect of the performance of question type classification (QT) and semantic phrase extraction (SPE) on the question generation (QG) task.

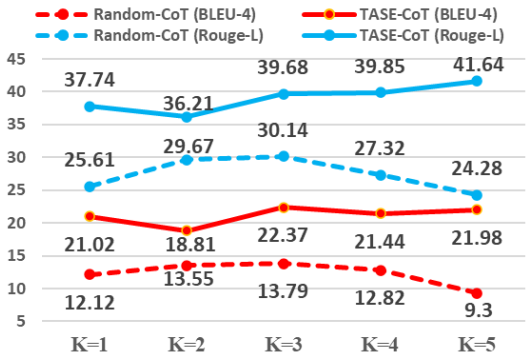

Figure 3: The figure shows the BLEU-4 and ROUGE-L scores of TASE-CoT and Random-CoT on HotpotQA with different numbers of demonstration examples in few-shot settings.

on different templates further demonstrate that our method is strongly robust to different templates.

4.4 Effect of Question Type and Semantics Phrase Extraction

We explore the effect of extraction tasks (i.e., question type extraction and semantics phrase extraction) on the few-shot MQG task. Particularly, we measure the accuracy of the question type classification and semantic phrase extraction on the testing set and report the results in Table 8. We observe that the performance of MQG increases with higher performance on question type classification and semantic phrase extraction. This confirms the motivation of our approach to leverage question types and semantic phrases to improve MQG.

4.5 Effect of Demonstrations Number

We investigate the influence of the number of demonstration examples (i.e., K) on the MQG. We try different numbers of K and report the corresponding model performance in Figure 3. From the curves, we observe that $K = 3$ is the number of demonstrations that achieve the best performance in both Random-CoT and TASE-CoT. However, the performance of Random-CoT drops significantly on both sides of $K = 3$. Compared with Random-CoT, the performance of our approach does not fluctuate much around $K = 3$, which shows that our few-shot approach is robust on the MQG task while only requiring a few demonstration examples to achieve good results.

Context:
[1] Walking with Elephants: It was one of ***the most popular songs*** *in* ***Ibiza*** *2014.*
[2] Ten Walls: Marijus Adomaitis (born 19 January 1983), better known by his stage names Ten Walls or Mario Basanov, is a ***Lithuanian producer*** *who is best* ***known*** *for his 2014 single "**Walking with Elephants**", which peaked at* ***number 6*** *on* ***the UK Singles Chart****.*
Answer: *Ten Walls*
Gold Question: ***What*** ***Lithuanian producer*** *is* ***known*** *for a song that was one of* ***the most popular songs*** *in* ***Ibiza*** *in 2014?*

Vanilla Prompt (fixed): ***Who*** *is the artist* ***known*** *for the 2014 single "**Walking with Elephants**"?*
Vanilla Prompt (varied): ***Who*** *is the* ***Lithuanian producer*** *best* ***known*** *for his 2014 single "**Walking with Elephants**"?*
Random-CoT (fixed): ***Who*** *is the* ***Lithuanian producer*** ***known*** *for his 2014 single "**Walking with Elephants**"?*
Random-CoT (varied): ***Who*** *is the* ***Lithuanian producer*** ***known*** *for his 2014 single "**Walking with Elephants**," which was one of* ***the most popular songs*** *in* ***Ibiza*** *2014?*
Manual-CoT: ***Who*** *is the* ***Lithuanian producer*** ***known*** *for the 2014 single "**Walking with Elephants**" that peaked at* ***number 6*** *on* ***the UK Singles Chart****?*
Auto-CoT: ***Which*** ***Lithuanian producer*** *is* ***known*** *for one of* ***the most popular songs*** *in* ***Ibiza*** *2014?*
TASE-CoT: ***What*** ***Lithuanian producer*** *is best* ***known*** *for a song that was one of* ***the most popular songs*** *in* ***Ibiza*** *in 2014?*

Figure 4: The figure shows the case study of one representative example from HotpotQA test set. We indicate the important and unimportant semantic phrases in red and blue colors, respectively. The appropriate and inappropriate question words are highlighted in light blue and green, respectively.

4.6 Case Study

In Figure 4, we present a case study to demonstrate the effectiveness of TASE-CoT on MQG, where the question generated by TASE-CoT and baselines are presented. The question generated by TASE-CoT shows more complexity compared with the baselines, in that our approach accurately selects the most appropriate question type and extracts the related semantic phrases accordingly. Both question type and essential semantic phrase contribute to improving task performance. In contrast, baselines easily extract less important semantic phrases owning to selecting the inappropriate question type, which hurts the performance of MQG.

5 Related Work

This paper is relevant to multi-hop question generation and CoT. The following text presents the details of the related work in the two fields.

Muti-hop Question Generation Early research on QG predominantly concentrated on generating

shallow factual questions based on a single document. Recently, researchers have shown an increasing interest in addressing the challenges of complex multi-hop question generation (MQG) tasks. However, the difficulty in generating multi-hop questions lies in selecting questioning information relevant to a given answer for questioning from multiple documents and using it as a foundation to generate questions in a manner consistent with human style. For this, many studies (Pan et al., 2020; Su et al., 2020; Fei et al., 2021) propose semantic graph-based methods, which aim to solve MQG by extracting semantics related to the answer from the context. To further enhance performance, some research Fei et al. (2022); Xia et al. (2023) explore the decoder-enhanced method based on the semantic graph-based method and achieve great performance. Our research is different from the existing ones as we mainly focus on solving few-shot MQG challenges with LLMs.

Chain-of-Thought CoT is an emerging prompting technique, which improves the performance of LLMs by instructing LLMs to produce intermediate reasoning steps in tasks. Consequently, with the rise of LLMs, diverse CoT prompting methods have been explored in current research. Kojima et al. (2023) initially introduce zero-shot-CoT using the prompt "*Let's think step by step.*". The Manual-CoT method, proposed by Wei et al. (2022), involves crafting human-written few-shot CoT demonstrations. Least-to-most prompting (Zhou et al., 2023) utilize problem decomposition to create a CoT prompt. A self-consistency decoding strategy is introduced by Wang et al. (2023) to sample diverse reasoning paths and choose the most consistent answer by marginalizing these paths. The Auto-CoT method (Zhang et al., 2022), automatically generates CoT demonstrations by leveraging LLMs. To the best of our knowledge, there are no existing studies that apply few-shot CoT to MQG and we are the first to do so.

6 Conclusion

In this paper, we propose TASE-CoT for the few-shot MQG. TASE-CoT extracts the question type and type-aware semantic phrases from the given documents and the answer, then utilizes them to conduct the question generation with type-aware CoT. We run experiments on benchmark datasets, and the results on benchmark datasets show that our approach achieves state-of-the-art performance.

Acknowledgements

This paper is supported by the National Key Research and Development Program of China under the grant (2021YFF0901600).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-shot Learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. In *Advances in Neural Information Processing Systems*, volume 32.

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, Vancouver, Canada.

Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. 2023. Reasoning Implicit Sentiment with Chain-of-Thought Prompting. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1171–1182, Toronto, Canada.

Zichu Fei, Qi Zhang, Tao Gui, Di Liang, Sirui Wang, Wei Wu, and Xuanjing 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, Dublin, Ireland.

Zichu Fei, Qi Zhang, and Yaqian Zhou. 2021. Iterative GNN-based Decoder for Question Generation.

In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2573–2582, Online and Punta Cana, Dominican Republic.

Ruyi Gan, Ziwei Wu, Renliang Sun, Junyu Lu, Xiaojun Wu, Dixiang Zhang, Kunhao Pan, Ping Yang, Qi Yang, Jiaxing Zhang, and Yan Song. 2023. Ziya2: Data-centric Learning is All LLMs Need. *arXiv preprint arXiv:2311.03301*.

Jialong Han, Yan Song, Wayne Xin Zhao, Shuming Shi, and Haisong Zhang. 2018. Hyperdoc2vec: Distributed Representations of Hypertext Documents. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2384–2394, Melbourne, Australia.

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, Barcelona, Spain (Online).

Antje Janssen, Davinia Rodríguez Cardona, Jens Passlick, and Michael H. Breitner. 2022. How to Make Chatbots Productive – A User-oriented Implementation Framework. *International Journal of Human-Computer Studies*, 168:102921.

Tianbo Ji, Chenyang Lyu, Zhichao Cao, and Peng Cheng. 2021. Multi-hop Question Generation Using Hierarchical Encoding-Decoding and Context Switch Mechanism. *Entropy*, 23(11).

Yanghoon Kim, Hwanhee Lee, Joongbo Shin, and Kyomin Jung. 2019. Improving Neural Question Generation Using Answer Separation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6602–6609.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large Language Models are Zero-Shot Reasoners. ArXiv:2205.11916 [cs].

Alon Lavie and Abhaya Agarwal. 2007. METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 228–231, Prague, Czech Republic.

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, Online.

Yehao Li, Yingwei Pan, Ting Yao, and Tao Mei. 2022. Comprehending and Ordering Semantics for Image Captioning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17969–17978.

Yuanyuan Liang, Jianing Wang, Hanlun Zhu, Lei Wang, Weining Qian, and Yunshi Lan. 2023. Prompting Large Language Models with Chain-of-Thought for Few-Shot Knowledge Base Question Generation.

Chin-Yew Lin. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain.

Bang Liu, Mingjun Zhao, Di Niu, Kunfeng Lai, Yancheng He, Haojie Wei, and Yu Xu. 2019. Learning to Generate Questions by LearningWhat not to generate. In *The World Wide Web Conference*, WWW '19, page 1106–1118, New York, NY, USA.

Nikahat Mulla and Prachi Gharpure. 2023. Automatic Question Generation: A Review of Methodologies, Datasets, Evaluation Metrics, and Applications. *Prog. in Artif. Intell.*, 12(1):1–32.

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, Online.

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, Philadelphia, Pennsylvania, USA.

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. *J. Mach. Learn. Res.*, 21(1).

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China.

Devendra Singh Sachan, Lingfei Wu, Mrinmaya Sachan, and William Hamilton. 2020. Stronger Transformers for Neural Multi-Hop Question Generation.

Damien Sileo, Tim Van De Cruys, Camille Pradel, and Philippe Muller. 2019. Mining Discourse Markers for Unsupervised Sentence Representation Learning. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3477–3486, Minneapolis, Minnesota.

Marita Skjuve, Asbjørn Følstad, Knut Inge Fostervold, and Petter Bae Brandtzaeg. 2022. A Longitudinal Study of Human–chatbot Relationships. *International Journal of Human-Computer Studies*, 168:102903.

Yan Song, Chia-Jung Lee, and Fei Xia. 2017. Learning Word Representations with Regularization from Prior Knowledge. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 143–152.

Yan Song and Shuming Shi. 2018. Complementary Learning of Word Embeddings. In *IJCAI*, pages 4368–4374.

Yan Song, Tong Zhang, Yonggang Wang, and Kai-Fu Lee. 2021. ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders. *arXiv preprint arXiv:2105.01279*.

Dan Su, Peng Xu, and Pascale Fung. 2022a. QA4QG: Using Question Answering to Constrain Multi-hop Question Generation. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8232–8236.

Dan Su, Yan Xu, Wenliang Dai, Ziwei Ji, Tiezheng Yu, and Pascale Fung. 2020. Multi-hop Question Generation with Graph Convolutional Network. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4636–4647, Online.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2022b. Selective Annotation Makes Language Models Better Few-Shot Learners.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. ArXiv:2203.11171 [cs].

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Zehua Xia, Qi Gou, Bowen Yu, Haiyang Yu, Fei Huang, Yongbin Li, and Cam-Tu Nguyen. 2023. Improving Question Generation with Multi-level Content Planning. ArXiv:2310.13512 [cs].

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. ArXiv:1809.09600 [cs].

Jianxing Yu, Qinliang Su, Xiaojun Quan, and Jian Yin. 2023. Multi-hop Reasoning Question Generation and Its Application. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):725–740.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic Chain of Thought Prompting in Large Language Models.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models. ArXiv:2205.10625 [cs].

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2018. Neural Question Generation from Text: A Preliminary Study. In *Natural Language Processing and Chinese Computing*, pages 662–671, Cham.