Modeling *What-to-ask* and *How-to-ask* for Answer-unaware Conversational Question Generation

Xuan Long Do[1,2,*], **Bowei Zou**[1], **Shafiq Joty**[2,3†], **Anh Tai Tran**[4],
Liangming Pan[5], **Nancy F. Chen**[1], **Ai Ti Aw**[1]
[1]Institute for Infocomm Research (I^{2}R), A*STAR, Singapore,
[2]Nanyang Technological University, Singapore, [3]Salesforce AI,
[4]ByteDance, [5]University of California, Santa Barbara
{doxuanlong15052000, anhtai2672000}@gmail.com,
liangmingpan@ucsb.edu, {zou_bowei, nfychen, aaiti}@i2r.a-star.edu.sg, srjoty@ntu.edu.sg

Abstract

Conversational Question Generation (CQG) is a critical task for machines to assist humans in fulfilling their information needs through conversations. The task is generally cast into two different settings: answer-aware and answer-unaware. While the former facilitates the models by exposing the expected answer, the latter is more realistic and receiving growing attentions recently. *What-to-ask* and *how-to-ask* are the two main challenges in the answer-unaware setting. To address the first challenge, existing methods mainly select sequential sentences in context as the rationales. We argue that the conversation generated using such naive heuristics may not be natural enough as in reality, the interlocutors often talk about the relevant contents that are not necessarily sequential in context. Additionally, previous methods decide the type of question (boolean/span-based) to be generated implicitly. Modeling the question type explicitly is crucial in this (answer-unaware) setting, as the answer which hints the models to generate a boolean or span-based question, is unavailable. To this end, we present *SG-CQG*, a two-stage CQG framework. For the *what-to-ask* stage, a sentence is selected as the rationale from a semantic graph that we construct, and extract the answer span from it. For the *how-to-ask* stage, a classifier determines the target answer type of the question via two explicit control signals before generating and filtering. In addition, we propose *Conv-Distinct*, a novel evaluation metric for CQG, to evaluate the diversity of the generated conversation from a context. Compared with the existing answer-unaware CQG models, the proposed *SG-CQG* achieves state-of-the-art performance.

1 Introduction

Building systems that can comprehend human speech and provide assistance to humans through conversations is one of the main objectives in AI. Asking questions during a conversation is a crucial conversational behavior that helps AI agents communicate with humans more effectively (Allen et al., 2007; Li et al., 2016b). This line of research is known as *Conversational Question Generation (CQG)*, which targets generating questions given the context and conversational history (Nakanishi et al., 2019; Pan et al., 2019a; Gu et al., 2021; Do et al., 2022). Compared to traditional single-turn question generation (Pan et al., 2019b), CQG is more challenging as the generated multi-turn questions in a conversation need not only to be coherent but also follow a naturally conversational flow.

Generally, there are two main settings for the CQG task: answer-aware and answer-unaware. In the answer-aware setting, the expected answers of the (to be) generated questions are exposed to the models (Gao et al., 2019; Gu et al., 2021; Shen et al., 2021; Do et al., 2022). In reality, however, the answers are only “future” information that are unknown beforehand. Thus, growing attention has been on the more realistic answer-unaware setting, in which the answers are unknown to the CQG model (Wang et al., 2018; Pan et al., 2019a; Nakanishi et al., 2019; Qi et al., 2020; Do et al., 2022).

Prior studies either attempt to ask the questions first, and compute the reward function to evaluate their answerability (Pan et al., 2019a) or informativeness (Qi et al., 2020); or they extract the answer spans from the context as the *what-to-ask* first, and generate the questions based on them (Nakanishi et al., 2019; Do et al., 2022). However, it has been argued that the former approach tends to generate repetitive questions (Qi et al., 2020; Do et al., 2022). For the latter approach, Do et al. (2022) recently proposed a selection module to shorten the context and history of the input and achieved state-of-the-art performance. Nonetheless, it simply employs a naive heuristic to select the earliest forward sentence (without traceback) in the context as the

[*] Contribution during the internship at Institute for Infocomm Research.

[†]Work done when the author was on leave from NTU.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 10785–10803
July 9-14, 2023 ©2023 Association for Computational Linguistics

rationale to extract the answer span. Although such heuristics ensure the flow of the generated questions is aligned with the context, we argue that the resulting conversations may not be natural enough, because, in reality, the interlocutors often talk about the relevant parts that may not form a sequential context. Furthermore, previous studies (Gao et al., 2019; Do et al., 2022) trained the models to decide the type of the question (boolean/span-based) to be generated implicitly. We argue that modeling question type explicitly is critical since in this setting, the answer, which hints the models to generate a boolean or span-based question, is unavailable.

To address the above problems, we propose a two-stage CQG framework based on a semantic graph, *SG-CQG*, which consists of two main components: *what-to-ask* and *how-to-ask*. In particular, given the referential context and dialog history, the *what-to-ask* module *(1)* constructs a semantic graph, which integrates the information of coreference, co-occurrence, and named entities from the context to capture the keyword chains for the possible "jumping" purpose; *(2)* traverses the graph to retrieve a relevant sentence as the rationale; and *(3)* extracts the expected answer span from the selected rationale (Section 3.1). Next, the *how-to-ask* module decides the question type (boolean/span-based) via two explicit control signals and conducts question generation and filtering (Section 3.2).

In order to exhaustively assess the quality of the generated question-answer pairs, we propose a set of metrics to measure the *diversity*, *dialog entailment*, *relevance*, *flexibility*, and *context coverage* through both standard and human evaluations. Compared with the existing answer-unaware CQG models, our proposed *SG-CQG* achieves state-of-the-art performance on the standard benchmark, namely the CoQA dataset (Reddy et al., 2019).

Our contributions can be summarized as follows:

(1) We propose *SG-CQG*, a two-stage framework, which consists of two novel modules: *what-to-ask* encourages the models to generate coherent conversations; and *how-to-ask* promotes generating naturally diverse questions. Our codes will be released at `https://github.com/dxlong2000/SG-CQG`.

(2) *SG-CQG* achieves state-of-the-art performance on answer-unaware CQG on CoQA.

(3) To the best of our knowledge, we are the first to propose a set of criteria to comprehensively evaluate the generated conversations. Moreover, we propose *Conv-Distinct* to measure the diversity of the generated conversation from a context, which takes the context coverage into account.

(4) We conduct thorough analysis and evaluation of the questions and answers of our generated conversations, which can bring some inspiration for future work on the answer-unaware CQG.

2 Related Work

Our work is closely related to two lines of prior work. Extended related work is in Appendix A.1.

2.1 Conversational Question Generation

Question Generation has gained much attention from the research community over the years (Pan et al., 2019b; Lu and Lu, 2021). Despite such intensive exploration, much less attention has been drawn to Conversational QG or CQG. Generally, CQG has been considered in two main settings: answer-aware and answer-unaware. In the answer-aware setting, the expected answers are revealed to models (Gao et al., 2019; Gu et al., 2021; Shen et al., 2021; Do et al., 2022). However, this is not always the case in reality, as the answers are "future information". The answer-unaware setting; therefore, receives growing interests recently (Wang et al., 2018; Pan et al., 2019a; Nakanishi et al., 2019; Qi et al., 2020; Do et al., 2022).

To tackle the *what-to-ask* problem, prior studies (Pan et al., 2019a; Do et al., 2022) selected the next sentence in the context as the rationale. Do et al. (2022) extract the target answer span from the rationale, while Pan et al. (2019a) generate the question, and compute a reward function to fine-tune the model by reinforcement learning. The *how-to-ask* challenge was simply formulated as that in the answer-aware setting. In contrast, we attempt to model the rationale selection in a more coherent way by constructing and traversing a semantic graph, which simulates the keyword chains. We further propose control signals to promote diversity and fluency in question generation.

2.2 Knowledge-grounded Conversation Generation

Leveraging graphs to enhance dialog response generation has received growing interest (Moghe et al., 2018; Liu et al., 2019b; Xu et al., 2020, 2021). In particular, Xu et al. (2020) proposed to extract event chains (Mostafazadeh et al., 2016), and utilised them to help determine a sketch of a multi-turn dialog. Nonetheless, the situation differs significantly when it comes to the CQG task. The responses in the dialog response generation task are normally full sentences with enough relevant mentions. However, in CQG, the questions and answers are mostly short and lack clear keywords, which makes the existing keyword-graph not applicable. We thus present a semantic graph, which incorporates the coreference, co-occurrence, and named entities information from the context.

3 *SG-CQG*

We formulate the answer-unaware conversational question generation (CQG) task as: given the referential context $C = \{s_1, s_2, ..., s_m\}$ with s_i being the i-th sentence in context, and the conversational history $H_n = \{(q_1, a_1), (q_2, a_2), ..., (q_{n-1}, a_{n-1})\}$ with (q_i, a_i) being the i-th turn of the question-answer pairs, as input $\mathcal{D}_n = \{C, H_n\}$, the model learns to generate the current question q_n and answer a_n.

Figure 1 demonstrates an overview of our proposed framework. It consists of two main components: *(1)* A *what-to-ask* module aims to select a reasonable sentence in the referential context C as the current rationale r_n and thereby a span in r_n as the target answer a_n, given $\mathcal{D}_n$. *(2)* A *how-to-ask* module aims to generate the question q_n, guided by the rationale r_n and target answer a_n.

3.1 *What-to-ask* Module (WTA)

Existing answer-unaware CQG models (Pan et al., 2019a; Do et al., 2022) commonly utilize the next sentence of r_{n-1} in the context as the current rationale r_n. Although such heuristics can guarantee that the flow of the generated questions is consistent with the narrative in context, the generated conversation may not always be as natural as in reality, since human speakers often jump back and forth across the relevant but not sequential contents in context. To facilitate the models in selecting the current rationale and target answer appropriately and further improve the semantic diversity of dialogue flow, we design a *what-to-ask* module, which consists of two components: *semantic graph construction* and *graph traversal algorithm*.

Semantic Graph Construction (SGC) Figure 1 shows an example of our semantic graph. Each node is displayed as a textual span and the index of the sentence it belongs to. To construct the semantic graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, we first obtain the coreference clusters from the context C by AllenNLP (Shi and Lin, 2019) and build the set of initial nodes from phrases in the clusters. We then connect all the nodes in the same cluster as a chain: each node in the cluster (except the one that appears last in the context) is connected to the nearest forward one in the context. We denote this type of relation as *Coreference*. To enhance the connectedness of $\mathcal{G}$, we extract all named entities by *spaCy*[1] and add them as additional nodes if they are not in any clusters. We then connect all the nodes in the same sentence in the context in the same chaining style and name those edges as *Same Sentence*. Finally, we add a type of *Extra* edges between all connected subgraphs to make $\mathcal{G}$ fully-connected. Since those *Extra* edges do not bring any semantic relation to the graph, our objective is to minimize the number of those edges. Specifically, we gradually select, and connect two sentences such that their nodes are in different connected components and have the smallest indexes with the smallest difference, until the graph is fully-connected. To connect two sentences, we add an *Extra* edge between the last phrase in the smaller-index sentence and the first phrase in the remaining sentence. The adding-*Extra*-edges algorithm is in Appendix A.4.

Graph Traversal Algorithm (GTA) Given the conversational history H_n and the semantic graph $\mathcal{G}$, we create a queue q to store nodes for traversing. We first add the nodes that appear in any previous turn' rationale to q in the index order [2]. We then traverse $\mathcal{G}$ by popping the nodes in q until it becomes empty. For each node, we retrieve the sentence that contains it as the rationale r_n. If the model can generate a valid question from r_n and any answer span extracted from r_n, we add all unvisited neighbors of the current node to the beginning of q. A question is considered being valid if it passes the QF module (Section 3.2). Prepending the neighbors to queue is to prioritize the nodes that are connected so that the generated conversation can be formed from a chain of relevant sentences, which consolidates the coherence of the conversation. If the model cannot generate any valid q_n by the current node, we add its unvisited neighbors to the end of q. The pseudocode of our proposed *Graph Traversal Algorithm* is described in Appendix A.2.

[1] https://spacy.io/

[2] The nodes are ordered according to their sentences' indexes in the original context.

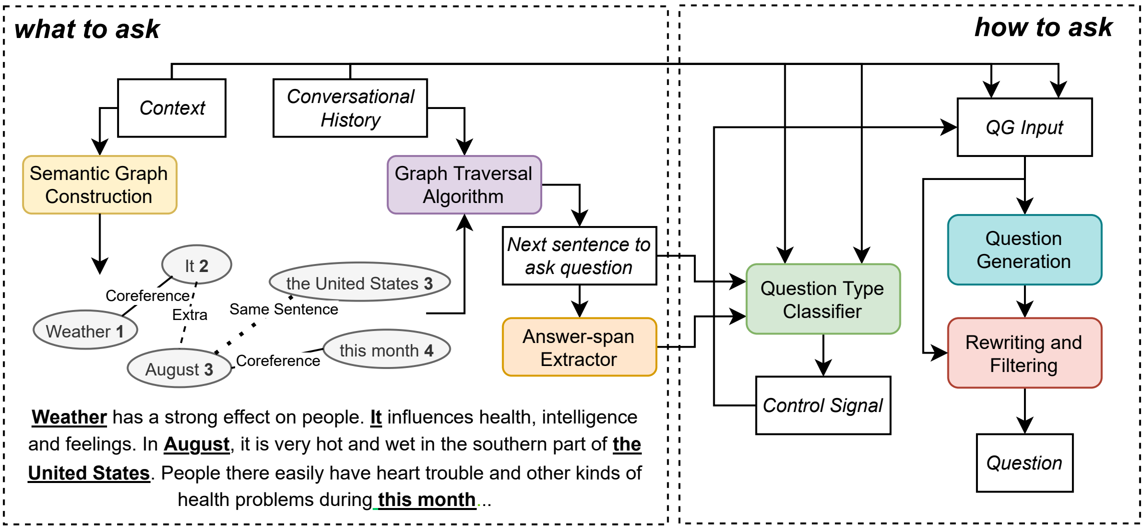
what to ask
how to ask
Context
Conversational History
Semantic Graph Construction
Graph Traversal Algorithm
Next sentence to ask question
Answer-span Extractor
It 2
the United States 3
Coreference
Extra
Same Sentence
Weather 1
August 3
Coreference
this month 4
Weather has a strong effect on people. It influences health, intelligence and feelings. In August, it is very hot and wet in the southern part of the United States. People there easily have heart trouble and other kinds of health problems during this month...
QG Input
Question Type Classifier
Control Signal
Question Generation
Rewriting and Filtering
Question

Figure 1: An overview of our proposed *SG-CQG* framework. It consists of two modules: the *what-to-ask* module aims to select a sentence as the rationale from the context and extracts the target answer span from it, and the *how-to-ask* module then predicts the type of the question to be generated and generates the question guided by that type.

Answer Span Extractor (AE) We follow Do et al. (2022) to design the answer span extractor module. In particular, a T5 model is trained on SQuAD (Rajpurkar et al., 2016) to predict the target answer span (a), given its original sentence in context (r). We use this pretrained model to extract a_n from r_n. Note that we also deselect the answer spans that are the same as those of previous turns.

Type	Example
Wrong answer	’Did he eat for breakfast?’, ’breakfast’
Irrelevant	’Was he still alive?’, ’no’,
Uninformative	’What happened one day?’, ‘Justin woke up very excited’, ’Who woke up?’, ‘Justine’
Redundant	’Did he eat something?’, ’yes’,...., ’Was he eating something?’, ’yes’

Table 1: Different types of common errors that CQG models are prone to without our extra postprocessing heuristics.

3.2 *How-to-ask* Module (HTA)

A high ratio of boolean questions in conversational datasets such as CoQA (Reddy et al., 2019) (around 20%) is one of the main challenges for current CQG studies (Gao et al., 2019; Pan et al., 2019a; Gu et al., 2021). To the best of our knowledge, however, there is no up-to-date work which attempts to tackle this challenge. This problem is even worse in the answer-unaware setting since there is no `Yes/No` answer to be provided to guide the generation of the models. Previous studies (Pan et al., 2019a; Do et al., 2022) simply train the CQG models to let them implicitly decide when to generate the boolean and span-based questions without any explicit modeling of the question type. We argue that explicitly modeling the question type is critical, as the models will gain more control on generating diverse questions, thus making the conversation become more natural. To this end, we introduce two control signals as the additional input to the QG model, and develop a simple mechanism to select the signal for the current turn.

Question Type Classifier (QTC) We design two control signals to guide the QG model: `<BOOLEAN>` is prepended to the textual input if we expect the model to generate a boolean question, and `<NORMAL>` otherwise. To classify which signal should be sent to the QG model, we train a RoBERTa (Liu et al., 2019a) as our *Question Type Classifier*. This binary clasifier takes the rationale r_n and the answer span a_n generated from *what-to-ask* module, the context and the shortened conversational history as the input, and generates the label $0/1$ corresponding to `<NORMAL>/<BOOLEAN>`. We conduct additional experiments to discuss why the *control_signals* work in Section 6.3.

Rewriting and Filtering (RF) Our RF module serves two purposes. Firstly, following Do et al. (2022), we train a T5 model on CoQA (Reddy et al., 2019) as our CQA model to answer the generated questions. A question is passed this filtering step if the answer generated by the CQA model has a fuzzy matching score greater or equal to 0.8 with the input answer span. Secondly, when invigilating the generated conversations, we observe multiple other errors that the blackbox model encounters, as shown in Table 1. We thus propose extra post-processing heuristics to filter out the generated questions and try to avoid the following issues: *(1) Wrong answer*. Unlike Do et al. (2022) that took the extracted spans as the conversational answers, we rewrite the extracted answer spans for the boolean questions by selecting the answers generated from the CQA model; *(2) Irrelevant*. For each generated question, we remove stopwords and question marks only for filtering purpose, and we check if all the remaining tokens exist in the context C; *(3) Uninformative*. To remove the turns like *("Who woke up?", "Justine")*, we check validity if no more than 50% of the tokens of r_n exist in any previously generated QA pairs; *(4) Redundant*. Unlike previous studies (Qi et al., 2020; Do et al., 2022) which only considered the redundant information from the generated answers, for each generated question that has more than 3 tokens, we filter it out if it has a fuzzy matching score >= 0.8 with any of the previously generated questions.

Question Generation (QG) We fine-tune a T5 model (Raffel et al., 2020) to generate conversational questions. We concatenate the input $\mathcal{D}_n^a = \{C, H_n, a_n, r_n, control\_signal\}$ in the format: `Signal:` $control\_signal$ `Answer:` a_n, r_n `Context:` C `[SEP]` H_{sub}, where $H_{sub} \in H_n$. The model then learns to generate the target question q_n. In our experiments, H_{sub} is the shortened H_n, in which we keep at most three previous turns. It was shown to improve upon training with the whole H_n significantly (Do et al., 2022). The performance of the QG model is in Appendix A.3.

4 Experimentation

4.1 Experimental Settings

Dataset We use CoQA (Reddy et al., 2019), a large-scale CQA dataset, in our experiments. Each conversation includes a referential context and multiple question-answer pairs, resulting in a total of 127k question-answer pairs. Among them, around 20% of questions are boolean, which makes this dataset become challenging for the CQG task (Pan et al., 2019a; Gu et al., 2021). Since the test set of CoQA is unavailable, we follow Do et al. (2022) to keep the original validation set as our *test set* and randomly sample 10% of the original training set as our new *validation set*.

Automatic Evaluation We utilise BERTScore (Zhang et al., 2020) as our dialog entailment metric (BERTScore-entailment), a generalization of Dziri et al. (2019). It considers the generated response (question/answer) as the premise, and the utterances in the conversational history as the hypothesis, and measures their similarity score as the topic coherence score. This property is crucial as the questions/answers should focus on the same topic as the previous turn(s). In our experiment, we measure the dialog entailment score with 1, 2, and all previous turn(s). To measure the relevance between the generated conversation and the context, we concatenate the generated QA pairs and compute the BERTScore. It provides how the generated conversation is explicitly relevant to the context.

We observe short conversations with very few generated turns tend to yield very high scores on the available diversity measurement metrics such as Distinct (Li et al., 2016a). Since the conversation is generated from a given context, we argue that how much information from the given context the generated conversation covers should be taken into account. To this end, we introduce *Context Coverage (CC)* to measure the percentage of the sentences in the context that are the rationales of generated QA pairs. Our proposed *Conv-Distinct* of a generated conversation is then computed by multiplying the Distinct score of the generated conversation with its CC score, to measure the diversity of the turns generated *from a given context*:

$$\textit{Conv-Distinct} = \textit{CC} * \text{Distinct} \qquad (1)$$

We further provide *Jumping Score* (JS) to measure the flexibility of the generated conversation. JS is defined as the percentage of turns in which the model jumps back to any previous content of their previous turn (i.e. trace-back). It is worth noting that we do not rank the models based on JS score. Details of proposed metrics are in Appendix A.7.

Human Evaluation Human evaluation is critical to evaluate the quality of the generated conversations since the CQG model may generate reasonable conversations but unmatched well with the provided ground-truth ones. We randomly select 25 contexts in our test set and take the first five generated turns from the output of each model to compare, resulting in 125 samples in total. We hire three annotators who are English native speakers. Each generated question is rated by annotators on a 1-3 scale (3 is the best). We follow Do et al. (2022) to utilize three criteria: **(1) Factuality** measures the factual correctness and meaning of generated questions, **(2) Conversational Alignment** measures how aligned the generated questions are with the

	Distinct		*Conv-Distinct*		BERTScore-entailment			BERTScore	*CC (%)*	*JS (%)*
Model	1	2	1	2	1	2	all			
BART	**84.09**	**97.25**	6.89	8.28	48.77	48.83	48.76	**82.07**	8.62	15.07
T5	60.31	82.20	14.44	19.77	77.51	79.23	77.01	81.13	23.33	13.83
GPT-2	60.12	88.06	19.72	26.99	77.77	79.70	77.18	79.49	34.50	7.50
CoHS-CQG	67.17	92.65	20.11	27.52	77.97	79.24	77.62	80.79	30.02	0.00
SG-CQG + w/o WTA	72.13	95.21	20.95	27.78	77.63	79.35	78.02	80.79	29.21	0.00
SG-CQG + w/o RF	21.00	50.01	21.00	50.01	80.55	81.16	78.13	77.74	100.00	6.69
SG-CQG + w/o QTC	57.47	91.28	38.93	62.13	81.95	83.20	79.18	80.76	68.06	19.67
SG-CQG (ours)	57.42	91.29	**38.99**†	**62.27**†	**81.99**†	**83.27**†	**79.29**†	80.89	**68.52**†	19.72
Oracle	58.29	80.10	33.60	52.89	81.93	82.95	79.36	81.05	58.10	16.11

Table 2: Performance of answer-unaware CQG on the test set (CoQA dev set). CC: Context Coverage score, JS: Jumping Score. † denotes our model significantly outperforms baselines with p-value < 0.01 under t-test (Appendix A.8).

	Distinct		*Conv-Distinct*		CC (%)
Model	1	2	1	2	
ReDR	22.15	33.42	-	-	-
T5	51.17	73.07	12.98	17.58	23.33
GPT-2	57.79	88.04	18.89	24.93	34.50
CoHS-CQG	66.18	90.01	19.05	25.67	30.02
SG-CQG + w/o WTA	**68.35**	**92.33**	19.66	26.47	29.21
SG-CQG + w/o RF	23.48	51.14	23.48	51.14	100.00
SG-CQG + w/o QTC	49.27	79.53	33.18	54.04	68.06
SG-CQG	54.15	79.61	**33.34**	**54.26**	**68.52**
Oracle	54.91	85.76	31.87	49.86	58.10

Table 3: Question generation evaluation results on our test set (CoQA validation set).

Model	EM (%)	F1 (%)	CC (%)
GPT-2	17.28	30.22	34.50
BART	18.64	38.23	8.62
T5	34.29	48.67	23.33
CoHS-CQG	35.14	52.08	30.02
SG-CQG + w/o WTA	38.89	56.17	29.21
SG-CQG + w/o RF	18.14	22.85	100.00
SG-CQG + w/o QTC	37.43	56.83	68.06
SG-CQG	**42.89**	**63.48**	**68.52**
Oracle	63.65	74.08	58.10

Table 4: Answer span extraction evaluation results on our test set (CoQA validation set).

history, **(3) Answerability** measures how answerable the generated questions are by the given context. Given the fact that LMs can generate fluent texts, we omit using *Fluency* and *Grammaticality*. We measure the annotators' agreement by Krippendorff's alpha (Krippendorff, 2011). Our human rating instructions are in Appendix A.9.

Implementation Details We fine-tune a RoBERTa$_{large}$ (Liu et al., 2019a) as our binary *Question Type Classifier* with the pretrained checkpoints from fairseq (Ott et al., 2019) on CoQA. We use a learning rate of 1e-5, a window size of 512, a batch size of 4, and AdamW (Loshchilov and Hutter, 2019) as our optimizer. Our classifier achieves an accuracy of 95.6%. The model is finetuned on a P40 Colab GPU for 10 epochs. Details of the input format are in Appendix A.5.

We initialise *SG-CQG* with pretrained checkpoints of T5$_{base}$ model (Raffel et al., 2020) from Huggingface (Wolf et al., 2020). We also use AdamW (Loshchilov and Hutter, 2019) as our optimizer with a warmup of 0.1 and an initial learning rate of 1e-4. We train the model for 100k iterations with a standard window size of 512, a batch size of 4, and use a Beam search decoding strategy with a beam size of 4.

5 Main Results

To evaluate the performance of SG-CQG on the answer-unaware CQG task, we employ 4 baselines for comparison, as shown in Table 2. *(1)* T5$_{base}$ (Raffel et al., 2020), *(2)* BART$_{base}$ (Lewis et al., 2020), *(3) GPT-2 (Radford et al., 2019)*, which are fine-tuned to generate conversational question-answer pairs end-to-end, and *(4)* CoHS-CQG (Do et al., 2022) which adopts a strategy to shorten the context and history of the input, achieves the SoTA performance on CoQA in answer-aware and answer-unaware CQG.

Firstly, we observe that SG-CQG outperforms other methods on most of the metrics, except Distinct and BERTScore. The reason is that BART and T5 often generate short QA pairs (the CC scores are 8.62% and 23.33% on average, respectively), and copy more from the context, thus they get higher scores on Distinct and BERTScore. Secondly, the metric Conv-Distinct reasonably penalizes models that generate too short conversations, on which SG-CQG achieves the best results. Thirdly, by allowing the model to jump back and forth across the relevant contents in the context by the semantic graph, SG-CQG outperforms other methods significantly on BERTScore-entailment, which indicates that conversational coherence is indeed improved. Furthermore, SG-CQG achieves the highest JS score, which demonstrates that the *what-to-ask* module allows our model to be most flexible in selecting rationales compared to the baselines. SG-CQG also achieves a significantly higher Context Coverage (CC) score compared to CoHS-CQG. Finally, compared with the results of Oracle, which are from the human-generated conversations, SG-CQG achieves commensurate performance on BERTScore-entailment and BERTScore. It demonstrates that our generated conversations are as closely coherent as human-generated ones.

Question Generation Evaluation We compare the generated conversational questions of our model with 4 baselines: *(1)* ReDR (Pan et al., 2019a) is an encoder-decoder framework which incorporates a reasoning procedure to better understand what has been asked and what to ask next about the passage; *(2)* T5$_{base}$ (Raffel et al., 2020); *(3) GPT-2 (Radford et al., 2019)*; *(4)* CoHS-CQG (Do et al., 2022). For T5, GPT-2 and CoHS-CQG, we extract the generated questions from the generated conversations for comparison. We measure the diversity of the generated questions by Distinct (Li et al., 2016a) and our proposed Conv-Distinct. Table 3 shows evaluation results of the generated conversational questions. We observe that *SG-CQG* achieves the best performance on Conv-Distinct, which takes the context coverage into account.

Answer Span Extraction Evaluation We further evaluate the generated conversational answers of our model with 4 baselines: *(1)* T5$_{base}$ (Raffel et al., 2020); *(2)* BART$_{base}$ (Lewis et al., 2020); *(3)* GPT-2 (Radford et al., 2019); *(4)* CoHS-CQG (Do et al., 2022). We extract the generated conversational answers from the generated conversations of the models for comparison. We train another T5$_{base}$ model on CoQA for the CQA task (see Appendix A.6) and utilize it to generate the *ground-truth* answers for the generated questions of the models. We then evaluate the quality of the generated conversational answers by measuring the *Exact Match (EM)* and *F1* scores with the *ground-truth* ones. Table 4 shows the evaluation results. We observe that the generated conversational answers extracted by *SG-CQG* achieve the best EM and F1 scores, which are significantly higher than the other baselines.

Human Evaluation The results of the human evaluation are present in Table 5. Generally, *SG-CQG* achieves the highest performances on all three proposed metrics with a good overall annotators' agreement with an alpha of 0.73. In particular, we observe that by integrating the semantic graph into the selection of the rationales, *SG-CQG* outperforms CoHS-CQG (Do et al., 2022) significantly in the conversational alignment property. Furthermore, *SG-CQG* improves CoHS-CQG by a gap in the answerability and factuality of the generated questions, which reflects that our RF module with additional post-processing steps works as expected.

Model	Fact.	C-Align	Ans.
T5	2.53	2.49	2.39
CoHS-CQG	2.54	2.52	2.46
SG-CQG	**2.61**	**2.62**	**2.53**
Krip.'s α	0.71	0.72	0.75

Table 5: Human evaluation results. *Fact.*: Factuality, *C-Align*: Conversational Alignment, *Ans.*: Answerability.

6 Discussion

6.1 Ablation Studies

Ablation of What-to-ask Module (WTA) To better understand how the *what-to-ask* module affects our proposed model in generating conversations, we study its ablation named *SG-CQG + w/o WTA* in Tables 2, 3, 4. In this case, our model becomes an upgraded version of CoHS-CQG (Do et al., 2022). Compared to CoHS-CQG, it achieves higher scores on all metrics except the Context Coverage (CC), which reflects that the quality of the generated conversations is indeed improved. These improvements are expected as the model in this case gains more control over generating boolean questions and has a stricter filtering process. This stricter filtering process also explains why it gets a lower CC score compared to CoHS-CQG.

Ablation of Question Type Classifier (QTC) We conduct an ablation study of the Question Type Classifier (QTC) module. We name this experiment *SG-CQG + w/o QTC*. Table 2 shows the evaluation results of generated question-answer pairs. Compared with SG-CQG, the performance of *SG-CQG + w/o QTC* drops slightly on nearly all metrics (except Distinct), which consolidates our hypothesis that explicitly modeling the question type improves the overall coherency of the conversation. Furthermore, Table 3 shows that QTC enhances the diversity of the generated questions, while Table 4 illustrates that QTC improves the quality of the

Context	Generated Conversation	Rationales
1. One day Mary took a walk to the park. 2. The park was very close to her house. 3. On her way to the park she passed her friend Kim's house. ... 7. John's house was three houses down. 8. Mary and Kim stopped by to ask John if he wanted to play at the park. ... 14. They loved the flowers and the swings! 15. Soon it was dinnertime and the girls went home.	Q1: What did Mary do? A1: Took a walk to the park Q2: Where did she see her friend? A2: Kim's house Q3: Who did they ask about going there? A3: John Q4: How far away was his home? A4: Three houses Q5: What time of day were they leaving? A5: Dinnertime Q6: Did they enjoy flowers? A6: Yes	1, 3, 8, 7, 15, 14

Table 6: One sample conversation generated by our model SG-CQG.

generated answers.

Ablation of Rewriting and Filtering (RF) *SG-CQG + w/o RF* in Table 2 shows the ablation results of the Rewriting and Filtering (RF) module. As removing the RF module means we do not filter out any generated question, it results in two consequences. Firstly, since for each sentence, the model can generate at least one conversational question, the CC score of *SG-CQG + w/o RF* is perfect (100%). Second, redundant questions and answers are generated very frequently. As such, removing the RF module reduces the quality of the generated question-answer pairs (Table 2) and questions (Table 3) significantly. Notably, without the RF module, the extracted answer spans by *SG-CQG + w/o RF* can be very different from the true conversational answers, resulting in very low F1 and EM scores (Table 4). Although the CC score is perfect, the generated question-answer pairs from this experiment are of bad-quality.

6.2 Case Study

We present one conversation generated by our proposed SG-CQG in Table 6. We observe that the rationale of Q2-A2 is the 3-rd sentence in the context, and the rationale of Q3-A3 is the 8-th sentence, which is a forward jump of the model. On the other hand, the rationale of the Q4-A4 is the 7-th sentence, which is a traceback. Such a traceback enhances reasonable coherence between Q3-A3 and Q4-A4. Furthermore, Q5-A5 to Q6-A6 is also a traceback, and especially, Q6 is a boolean question. More case studies are shown in Appendix A.10.

6.3 Why Do Control Signals Work?

Experimental Settings We design the experiments to verify the helpfulness of our two proposed *control_signals*: `<BOOLEAN>` and `<NORMAL>`. In particular, we train a T5 model (Raffel et al., 2020) in the answer-aware setting. Given the input $\mathcal{D}_n^a = \{C, H_n, a_n, r_n\}$ with C, H_n, a_n, r_n as the context, ground-truth conversational history, ground-truth answer, and round-truth rationale, respectively, we conduct three experiments in Table 9: original input with `Yes/No` keyword (*With* `Y/N`), original input without `Yes/No` keyword (*W/o* `Y/N`), original input without `Yes/No` and with the ground-truth *control_signal* (*W/o* `Y/N` + *control_signal*). Note that we train the model with the whole context, and a maximum of three previous history turns, as discussed in Appendix A.3. We measure the performance of the answer-aware CQG model separately on two types of questions: boolean and span-based by ROUGE-L (Lin, 2004) and BERTScore (Zhang et al., 2020).

Observations Table 9 shows the experimental results. We derive two main observations. Firstly, without knowing the keyword `Yes/No` (*W/o* `Y/N`) *- this is the case in the answer-unaware setting*, the model performs worse. This decrease shows that the `Yes/No` keyword is indeed helpful in hinting the model towards generating the correct questions. Secondly, by inputting the ground-truth *control_signal* into the model (*W/o* `Y/N` + *control_signal*), the performance is improved by a large margin compared to (*W/o* `Y/N`). We obtain three implications from the above improvement. Firstly, it consolidates our hypothesis that inputting the ground-truth *control_signal* is truly helpful. Secondly, by training with the *control_signal*, the performance of the model is even higher than with `Y/N` in the span-based cases, which indicates that training the model with *control_signal* makes it more stable to generate the correct questions. Thirdly, the performance of (*W/o* `Y/N` + *control_signal*) is lower than (*With* `Y/N`) in boolean cases. The reason is <BOOLEAN> only informs the model to generate a boolean question without informing to generate an Yes or No one.

7 Conclusion

This paper presents SG-CQG, a two-stage framework for the CQG task in the answer-unaware setting. Firstly, the *what-to-ask* module aims to select a sentence as the rationale by the proposed semantic graph and extract the answer span from it. The *how-to-ask* module classifies the type of the question before generating and filtering it. Additionally, we propose a set of automatic evaluation criteria for answer-unaware CQG, especially a novel metric, *Conv-Distinct*, to evaluate the generated conversation from a context. Extensive automatic evaluation and human evaluation show that our method achieves state-of-the-art performances in the answer-unaware setting on CoQA, with a significant improvement in the conversational alignment property compared to previous frameworks. In the future, we will focus on how to reason over our semantic graph to select the rationale, and further improve the performances of how-to-ask module.

Limitations

A limitation of our work is that our Graph Traversal Algorithm (Section 3.1) is a heuristic and unlearned algorithm. This leads to a number of nodes after being selected by this algorithm are not suitable for the model to generate conversational questions, and are eventually filtered out by other modules. Future works can focus on more advanced techniques to guide the model to select the nodes such as Graph Neural Networks (Wu et al., 2020). Furthermore, our algorithm to select the relevant turns in the conversational history to generate the conversational questions is a heuristic of selecting a maximum of three previous turns. This heuristic may not be optimal for the model to gather necessary information from history to generate conversational questions in the next turns, as discussed by Do et al. (2022).

Ethical Considerations

In this paper, we present a two-stage CQG framework (SG-CQG), which was trained on CoQA (Reddy et al., 2019), a published large-scale dataset for building Conversational Question Answering systems. Our framework is potentially helpful for building chatbot systems, which can serve different streams such as educational, medical, or commercial purposes.

Through human evaluations, we observe that our proposed method does not generate any discriminatory, insulting responses (questions and answers). We validate the proposed method and baseline models on human evaluation which involves manual labor. We hire three annotators to score 125 generated questions in total. The hourly pay is set to S$15, which is higher than the local statutory minimum wage. Therefore, we do not anticipate any major ethical concerns.

Acknowledgements

This research has been supported by the Institute for Infocomm Research of A*STAR (CR-2021-001). We would like to thank anonymous reviewers from ARR for their valuable feedback which helped us to improve our paper. We also want to thank Dr. Richeng Duan (A*STAR) for his feedback in the initial stage of the project.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, et al. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

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, Florence, Italy. Association for Computational Linguistics.

James Allen, Nathanael Chambers, George Ferguson, Lucian Galescu, Hyuckchul Jung, Mary Swift, and William Taysom. 2007. Plow: A collaborative task learning agent. In *Proceedings of the 22nd National Conference on Artificial Intelligence - Volume 2*, AAAI'07, page 1514–1519. AAAI Press.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Xuan Long Do, Bowei Zou, Liangming Pan, Nancy F. Chen, Shafiq Joty, and Ai Ti Aw. 2022. CoHS-CQG: Context and history selection for conversational question generation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 580–591, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Nouha Dziri, Ehsan Kamalloo, Kory Mathewson, and Osmar Zaiane. 2019. Evaluating coherence in dialogue systems using entailment. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 146–148, Florence, Italy. Association for Computational Linguistics.

Yifan Gao, Piji Li, Irwin King, and Michael R. Lyu. 2019. Interconnected question generation with coreference alignment and conversation flow modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4853–4862, Florence, Italy. Association for Computational Linguistics.

Sarik Ghazarian, Johnny Wei, Aram Galstyan, and Nanyun Peng. 2019. Better automatic evaluation of open-domain dialogue systems with contextualized embeddings. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 82–89, Minneapolis, Minnesota. Association for Computational Linguistics.

Jing Gu, Mostafa Mirshekari, Zhou Yu, and Aaron Sisto. 2021. ChainCQG: Flow-aware conversational question generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2061–2070, Online. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability. *Computing*, 1.

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. Association for Computational Linguistics.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016a. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Jiwei Li, Alexander H Miller, Sumit Chopra, Marc'Aurelio Ranzato, and Jason Weston. 2016b. Learning through dialogue interactions by asking questions. *International Conference on Learning Representations 2017*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Zhibin Liu, Zheng-Yu Niu, Hua Wu, and Haifeng Wang. 2019b. Knowledge aware conversation generation with explainable reasoning over augmented graphs. 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 1782–1792, Hong Kong, China. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Chao-Yi Lu and Sin-En Lu. 2021. A survey of approaches to automatic question generation:from 2019 to early 2021. In *Proceedings of the 33rd Conference on Computational Linguistics and Speech Processing (ROCLING 2021)*, pages 151–162, Taoyuan, Taiwan. The Association for Computational Linguistics and Chinese Language Processing (ACLCLP).

Nikita Moghe, Siddhartha Arora, Suman Banerjee, and Mitesh M. Khapra. 2018. Towards exploiting background knowledge for building conversation systems. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2322–2332, Brussels, Belgium. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Mao Nakanishi, Tetsunori Kobayashi, and Yoshihiko Hayashi. 2019. Towards answer-unaware conversational question generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering*, pages 63–71, Hong Kong, China. Association for Computational Linguistics.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Boyuan Pan, Hao Li, Ziyu Yao, Deng Cai, and Huan Sun. 2019a. Reinforced dynamic reasoning for conversational question generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2124, Florence, Italy. Association for Computational Linguistics.

Liangming Pan, Wenqiang Lei, Tat-Seng Chua, and Min-Yen Kan. 2019b. Recent advances in neural question generation.

Bo Pang, Erik Nijkamp, Wenjuan Han, Linqi Zhou, Yixian Liu, and Kewei Tu. 2020. Towards holistic and automatic evaluation of open-domain dialogue generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3619–3629, Online. Association for Computational Linguistics.

Raul Puri, Ryan Spring, Mohammad Shoeybi, Mostofa Patwary, and Bryan Catanzaro. 2020. Training question answering models from synthetic data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5811–5826, Online. Association for Computational Linguistics.

Peng Qi, Yuhao Zhang, and Christopher D. Manning. 2020. Stay hungry, stay focused: Generating informative and specific questions in information-seeking conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 25–40, Online. Association for Computational Linguistics.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

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. *Journal of Machine Learning Research*, 21(140):1–67.

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, Austin, Texas. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Siamak Shakeri, Cicero Nogueira dos Santos, Henghui Zhu, Patrick Ng, Feng Nan, Zhiguo Wang, Ramesh Nallapati, and Bing Xiang. 2020. End-to-end synthetic data generation for domain adaptation of question answering systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5445–5460, Online. Association for Computational Linguistics.

Lei Shen, Fandong Meng, Jinchao Zhang, Yang Feng, and Jie Zhou. 2021. GTM: A generative triple-wise model for conversational question generation. 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 3495–3506, Online. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling.

Yansen Wang, Chenyi Liu, Minlie Huang, and Liqiang Nie. 2018. Learning to ask questions in open-domain conversational systems with typed decoders. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2193–2203, Melbourne, Australia. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24.

Jun Xu, Zeyang Lei, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. 2020. Enhancing dialog coherence with event graph grounded content planning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3941–3947. International Joint Conferences on Artificial Intelligence Organization. Main track.

Jun Xu, Zeyang Lei, Haifeng Wang, Zheng-Yu Niu, Hua Wu, and Wanxiang Che. 2021. Discovering dialog structure graph for coherent dialog generation. 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 1726–1739, Online. Association for Computational Linguistics.

Yi-Ting Yeh, Maxine Eskenazi, and Shikib Mehri. 2021. A comprehensive assessment of dialog evaluation metrics. In *The First Workshop on Evaluations and Assessments of Neural Conversation Systems*, pages 15–33, Online. Association for Computational Linguistics.

Xiang Yue, Ziyu Yao, and Huan Sun. 2022. Synthetic question value estimation for domain adaptation of question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1340–1351, Dublin, Ireland. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Appendix

A.1 Extended Related Work

Our work is related to two more lines of prior work.

A.1.1 Synthetic Question-Answering (QA) Generation

Synthetic QA generation based on pretrained language models (LM) has been studied and demonstrated the helpfulness in improving the downstream Reading Comprehension (RC) task (Alberti et al., 2019; Puri et al., 2020; Shakeri et al., 2020). Alberti et al. (2019) proposed a novel method to generate synthetic data by combining models of question generation and answer extraction, and by filtering the results to ensure roundtrip consistency. However, this work differs from ours since it only considered the task of single-turn QA generation and focused only on extractive QA generation while we focus on multi-turn QA generation and have both span-based and boolean questions. Regarding the filtering technique, this work and Puri et al. (2020) used *round-trip filtering* method, which is similar to Do et al. (2022) and is a relaxed version of our filtering module. Shakeri et al. (2020) later introduced an end-to-end framework to generate QA data. This work used *LM filtering* method, which is similar to *sample-and-reranking* (Holtzman et al., 2020) and ours. In our case (as discussed in *(1) Wrong answer* error in Section 3.2), to filter QA pairs, we also sample multiple answers from a QA model and select the answers with the highest frequency and confidence score by the model. If the highest frequency one is different from the highest confidence one, we filter our the question.

A.1.2 Dialog Generation Evaluation

Dialog evaluation metrics have been studied extensively (Yeh et al., 2021). However, it is worth noting that this task is different from ours, since we prefer evaluating the questions in QA conversations only. In addition, when conducting experiments with reference-free dialog generation metrics like BERT-RUBER (Ghazarian et al., 2019) and HolisticEval (Pang et al., 2020), we observe that these metrics are not suitable for evaluating QA pairs since the questions and answers in QA conversations are normally shorter without many referential details among turns compared to dialog responses.

Previous works (Alberti et al., 2019; Puri et al., 2020; Shakeri et al., 2020) usually evaluated the generated QA data by training the RC systems with it and examining whether the synthetic data improves the RC systems without actually examining the synthetic data. Recent work (Do et al., 2022) evaluated the QA pairs manually. In addition, (Yue et al., 2022) proposed *question value estimator*, a novel module to estimate the usefulness of synthetic questions to improve the target-domain QA performance. However, this is not directly relevant to ours since even though the metric can evaluate the usefulness of the generated questions, it does not offer the actual properties of the generated questions. To the best of our knowledge, our work is the first one that proposes a set of criteria to evaluate the question-answer pairs in QA conversations. The performance of models evaluated by our proposed automatic evaluation metrics (Table 2) is positively correlated with human evaluation (Table 5) where we observe that improvements on our metrics are also improvements on human evaluation metrics.

A.2 Graph Traversal Algorithm

We present the pseudocode of our *Graph Traversal Algorithm*, which is described in Section 3.1.

Algorithm 1: Graph Traversal Algorithm

```
Input: $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$,
       $H = \{(q_1, a_1), ..., (q_{n-1}, a_{n-1})\}/\emptyset$.
Output: Index of the sentence as $r_n$
Initialize: $I$: nodes in rationales of $H$,
  $q$: queue of nodes to visit,
  Add nodes in $I$ to $q$ in the index order.
while $q$ is not empty do
    $cur$ = $q[0]$
    del $q[0]$
    if $cur$ is visited twice then
        continue
    end
    $r_n$ = retrieve sentence contains $q[0]$
    $A_n$ = answer spans set extracted from $r_n$
    if successfully generate $q_n$ from $r_n$ and
      any $a_n \in A$ then
        Add unvisited neighbors of $cur$ to
          the beginning of $q$
    else
        Add unvisited neighbors of $cur$ to
          the end of $q$
    end
end
```

#Pre. turns	ROUGE-L	BLEU-4	BERTScore
1	48.64	17.93	93.42
2	48.77	**18.27**	93.43
3	**48.84**	18.18	**93.46**
4	48.27	18.16	93.38
Full history	45.93	17.11	93.09

Table 7: Performance of the T5 model, training with different fixed number of previous turns on our validation set.

A.3 Question Generation

Given the input $\mathcal{D}_n^a = \{C, H_n, a_n, r_n, control\_signal\}$ in which C, H_n, a_n, r_n, $control\_signal$ are the context, conversational history, expected answer, rationale, and the control signal respectively, we fine-tune a T5$_{base}$ model (Raffel et al., 2020) as our question generation model. Do et al. (2022) showed that by training the T5 model with the whole context and the shortened conversational history, the performance of the model is improved. We replicate this experiment by reporting the performance of the T5 model with a different number of the previous history turns in Table 7. We derive the same observation as Do et al. (2022), which is the model performs the best with a maximum of two or three conversational previous turns. As such, we opt for selecting at most 3 previous turns to train our QG model.

A.4 Adding Extra Edges Algorithm

We provide the pseudocode for the adding-*Extra*-edges algorithm in Algorithm 2.

A.5 Details of Question Type Classifier

In this section, we detail our setting to train and validate the proposed *Question Type Classifier*. We conduct our experiments on train set, our test set (i.e. CoQA validation set) and our validation set of CoQA (Reddy et al., 2019). For each conversation, we automatically label its questions according to their answers. In particular, a question is labeled as boolean if its answer begins with `Yes/No/yes/no/YES/NO`, and span-based otherwise. Given the input $\mathcal{D}_n^a = \{C, H_n, a_n, r_n\}$ with C, H_n, a_n, r_n are the context, ground-truth conversational history, ground-truth answer, round-truth rationale respectively, we construct the input to the classifier as followed. If $a_n \in$ {`Yes, No, yes, no, YES, NO`}, the input to the classifier is `Answer:` r_n r_n `Context:` C `[SEP]` H_{sub}, else, the input is `Answer:` a_n r_n r_n `Context:` C `[SEP]` where H_{sub} is the short-

Algorithm 2: Adding Extra Edges

```
Input: $\mathcal{G} = \{(u, v)\}$ for u, v are nodes in
       directed graph that belong to the
       same sentence. For different
       sentences, only consider the starting
       node and the ending node.
Output: The set of newly added edges
Initialize: A disjoint set union (DSU) for
            checking whether 2 sentences
            are in the same component.
$addedEdges$ = []
$pairs$ = all pairs of 2 sentences
sort($pairs$) // for prioritizing
    those pairs with the
    minimum index difference
for $pair$ in $pairs$ do
    $p_1$, $p_2$ = $pair$[0], $pair$[1]
    $sameComponent$ = check the
      connectivity of $p_1$, $p_2$ by DSU
    if not $sameComponent$ then
        merge sentences $p_1$ and $p_2$ into the
          same component by DSU
        add new edge between the ending
          node of sentence $p_1$ with starting
          node of sentence $p_2$ to
          $addedEdges$
    end
end
return $addedEdges$
```

ened H_n, in which we keep at most three previous turns, and the output is `0/1` indicating whether the ground-truth question is boolean/span-based. Our classifier achieves an accuracy of 95.6%.

A.6 Details of CQA Model

We fine-tuned a T5 (Raffel et al., 2020) as our Conversational Question Answering (CQA) model on CoQA (Reddy et al., 2019). The input to the model follows the format: `Question: Q [SEP] Context: C [SEP] H_sub` in which `Q, C` are the question and the context respectively, and `H_sub` is the shortened conversational history with a maximum of 3 previous turns. Our CQA model achieves 63.65% Exact Match (EM) and 74.08% F1, as we presented in Table 4.

A.7 Evaluation Metrics Discussion

One of our core contributions is the set of criteria to evaluate question-answer conversations. In this section, we detail our intuitions as well as computations of the metrics.

A.7.1 Distinct-N (Li et al., 2016a)

Distinct-N (Li et al., 2016a) is a N-gram metric to measure the diversity of a sentence. In our experiments, we calculate Distinct-1 score and Distinct-2 score provided by Li et al. (2016a)[3].

A.7.2 Context Coverage and Conv-Distinct

As we discussed in Section 4, one critical shortcoming when directly applying Distinct-N to evaluate the QA conversations is that the conversations with very few turns tend to attain very high Distinct-N scores. To address this challenge, we introduce Context Coverage (CC) and Conv-Distinct.

Context Coverage (CC) is measured as the percentage of sentences that are rationales. For example, given a context of 6 sentences, among them, 5 sentences are selected as rationales for a generated conversation. Then the CC score of this generated conversation is 5/6 = 0.84.

To compute CC Scores for E2E models, we classify a sentence as a rationale if there is at least one question-answer pair generated from that sentence. As a result, the model of Do et al. (2022) and our *SG-CQG* can output which sentence is a rationale, and it is straightforward to compute the CC scores. However, the end-to-end outputs of BART (Lewis et al., 2020) and T5 (Raffel et al., 2020) are the question-answer pairs only, it is needed to find which sentence is a rationale of each pair. To do so, we adopt a simple heuristic. For each generated question-answer pair, we classify a sentence as its rationale if that sentence has the *longest common substring* with the concatenation of its question and answer among all the sentences in the context. By that, we get the set of sentences that are rationales.

Conv-Distinct is defined as the multiplication of the Distinct score of the generated conversation with its CC score. For example, in the above generated conversation, the Distinct-1 score is 60.50. So its Conv-Distinct-1 score is 60.50 * 0.84 = 50.42.

It is worth noting that the diversity in token level is a common property of the dialog which has been discussed in many previous works (Qi et al., 2020; Pang et al., 2020; Adiwardana et al., 2020).

A.7.3 BERTScore−entailment $1, 2, 3$

BERTScore−entailment is an upgraded version of Dialog−entailment metric (Dziri et al., 2019), which measures the topic coherence property by deep contextual representation. We follow Dziri et al. (2019) to characterize the consistency of dialogue systems as a natural language inference (NLI) problem (Dagan et al., 2006). This property is important for questions and answers in the QA conversation, because the questions should focus on the topic of previous turns, and the answers should focus on their questions. In our experiments, we compute BERTScore−entailment with $1, 2$, and all previous turn(s). The BERTScore calculation is adopted from its authors[4].

A.7.4 BERTScore (Zhang et al., 2020)

We observe that Distinct-N, Conv-Distinct-N and BERTScore-entailment are only to measure the quality of the QA pairs. None of them measures the relationship between QA pairs and the given context. As such, we propose to use BERTScore (Zhang et al., 2020) to measure the similarity of the generated conversation and the given context. It is worth noting that this metric only serves for measuring the similarity between the generated conversation and context only. A generated conversation with a very high similarity score with the given context does not reflect that it is a very good conversation, as in the case of BART (Lewis et al., 2020) in Table 2. We provide this metric to give

[3] https://github.com/neural-dialogue-metrics/Distinct-N

[4] https://github.com/Tiiiger/bert_score

audiences "a sense" of how the generated conversation is explicitly relevant to the given context.

A.7.5 EM & F1 Answerability Measurements

The Exact Match (EM) and F1 measurements in Section 5 are to evaluate the answerability and the correctness of our generated questions and answers respectively (i.e. the quality of the generated conversational answers). Since from a context, multiple conversations can be generated, we argue that one critical aspect of a good conversation is the quality of the generated conversational answers, i.e. the conversational questions must be answerable by the given context, and their answers must be exactly the generated conversational answers.

A.7.6 Jumping Score (JS)

To further understand the characteristics of each model in generating conversations, we measure its jumping score. We define this score as the percentage of turns in which the model jumps back to any previous content of their previous turn (i.e. trace-back). For example, a generated conversation with the indexes of rationales [1,4,3,5,8,6] has the JS score is 2/5 = 0.4. It has 2 turns (over a maximum of 5 jumping back turns) in which the model jumps back, which are the 3−rd turn and 6−th turn. It is worth noting that the JS only shows one of the aspects of the result analysis. We could not say a system with the highest JS is better than others. JS only reflects a kind of flexibility for a what-to-ask module to some extent. We observe that our proposed SG-CQG achieves the highest JS score, which reflects that our proposed *what-to-ask* module is the most flexible in terms of selecting the sentences in the context.

A.8 Statistical Significance of Results

We compute the Student's t-test to measure the significant difference between our model's performance and the best baseline for each evaluation metric with the null hypothesis `H0: There is no significant difference`, and `H1: There is a significant difference`. We obtained the p-values as in Table 2:

- Compared to T5: 4.32e-11 (BERT-entailment all), 5.20e-98, (BERT-entailment 1), 2.48e-34 (BERT-entailment 2).
- Compared to CoHS-CQG: 7.62e-188 (CC Score), 5.12e-119 (Conv-Distinct 1), 8.11e-173 (Conv-Distinct 2). The p-values, in this case, are too small because the improvements are intuitively significant.

Criterion	Scoring Rules
Factuality	**Score 1:** The generated question has no meaning or is factually wrong with the context. **Score 2:** The generated question is fluent, but has a minor grammatical error. **Score 3:** The generated question is factually correct and grammatically correct with the context.
Conversational Alignment	**Score 1:** The generated question is totally irrelevant to the conversation history. **Score 2:** The generated question is partially aligned to the conversation history. **Score 3:** The generated question is fully aligned with the conversation history.
Answerability	**Score 1:** The generated question is not answerable by the context. **Score 2:** The generated question is answerable by the context, but does not have the answer as the target answer (the target answer is extracted from the Answer-span extractor). **Score 3:** The generated question is answerable by the context and its answer is the target answer.

Figure 2: Human Rating System

We observe that all the p-values are less than .01, which indicates that our improvements on those metrics are significant.

A.9 Human Evaluation Scoring System

We describe how we instructed three annotators to point the generated questions based on three criteria: *Factuality*, *Conversational Alignment*, and *Answerability*, as discussed in Section 4, in Figure 2. Following the discussion of Do et al. (2022), it is noted that in the answer-unaware setting, there is no target answer and rationale. However, since the what-to-ask module first seeks for one sentence in the context as the rationale and extracts the target answer from it, we still have the target answer and rationale. We thus define the *Score 2, 3* based on the retrieved rationale and extracted target answer.

A.10 Extended Case Studies

We present additional samples generated by SG-CQG in Table 8. In these samples, the first turn of the conversations is input to the model, and it generates their following turns.

A.11 Supplement: Why Do Control Signals Work?

In this section, we supplement the experimental results of the experiments with the *control_signal*. The results are presented in Table 9, and the discussions are in Section 6.3.

Context	**Generated Conversation**	**Rationales**
1. Once there was a beautiful fish named Asta. 2. Asta lived in the ocean. 3. There were lots of other fish in the ocean where Asta lived. 4. They played all day long. 5. One day, a bottle floated by over the heads of Asta and his friends. 6. They looked up and saw the bottle. ... 10. "It looks like a bird's belly," said Asta. ... 14. They wanted to see what was inside. ... 18. The note was written in orange crayon on white paper. ... 26. This note is from a little girl. 27. She wants to be your friend. ...	Q1: What was the fish named? A1: Asta Q2: In what body of water? A2: The ocean Q3: Who was in it? A3: Asta and his friends Q4: What did they find? A4: A bottle Q5: Why did they discover it? A5: Because they wanted to see what was inside Q6: What part of the bottle was it? A6: It was a bird's belly Q7: What was the note written in? A7: Orange crayon Q8: On what? A8: White paper Q9: What did the little girl want? A9: She wants to be your friend	1, 2, 5, 6, 14, 10, 19, 19, 27
1. This is the story of a young girl and her dog. 2. The young girl and her dog set out a trip into the woods one day. 3. Upon entering the woods the girl and her dog found that the woods were dark and cold. 4. The girl was a little scared and was thinking of turning back, but yet they went on. 5. The girl's dog was acting very interested in what was in the bushes up ahead. 6. To both the girl and the dog's surprise, there was a small brown bear resting in the bushes. 7. The bear was not surprised and did not seem at all interested in the girl and her dog. 8. The bear looked up at the girl and it was almost as if he was smiling at her. ... 10. The girl and the dog kept walking and finally made it out of the woods. ...	Q1: What is the story of? A1: A young girl and her dog Q2: What was the weather like? A2: Dark and cold Q3: What did they go on? A3: A trip into the woods Q4: What kind of animal did they find? A4: A small brown bear Q5: How did it make them feel? A5: Scared Q6: How did they get out? A6: Kept walking Q7: Did they have a dog? A7: Yes Q8: How did the bear at her? A8: Smiling	1, 3, 2, 6, 4, 10, 1, 8

Table 8: Additional sample conversations generated by our model SG-CQG. The first turn of both conversations is given to the model.

	ROUGE-L (boolean/span-based)			BERTScore (boolean/span-based)		
Model	Precision (%)	Recall (%)	F1 (%)	Precision (%)	Recall (%)	F1 (%)
With Y/N	**38.70**/51.81	**38.97**/53.06	**37.73**/50.65	**93.12/93.66**	**92.96/93.90**	**93.03/93.77**
W/o Y/N	35.92/51.81	35.49/53.07	34.59/50.64	92.70/**93.66**	92.47/**93.90**	92.57/**93.77**
W/o Y/N + control_signal	37.56/**51.86**	37.18/**53.09**	36.22/**50.68**	92.96/**93.66**	92.74/**93.90**	92.84/**93.77**

Table 9: Performance of T5 model in different settings. Y/N denotes Yes/No keyword.

ACL 2023 Responsible NLP Checklist

A For every submission:

□ A1. Did you describe the limitations of your work?
Left blank.

□ A2. Did you discuss any potential risks of your work?
Left blank.

□ A3. Do the abstract and introduction summarize the paper's main claims?
Left blank.

□ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B □ Did you use or create scientific artifacts?

Left blank.

□ B1. Did you cite the creators of artifacts you used?
Left blank.

□ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Left blank.

□ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Left blank.

□ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Left blank.

□ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Left blank.

□ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Left blank.

C □ Did you run computational experiments?

Left blank.

□ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Left blank.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

□ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Left blank.

□ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Left blank.

□ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Left blank.

D □ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Left blank.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Left blank.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Left blank.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Left blank.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Left blank.