Multi-Source Pointer Network with Coverage Mechanism for Conversational Question Generation

Xiang Wu, Cai Guo, Hong-Ning Dai, *Senior Member, IEEE*

***Abstract*—In this paper, we study the problem of Conversational Question Generation (CQG). Different from the standalone interaction in general Question Generation (QG), CQG is grounded in a passage and combines conversation history to generate interconnected questions in a question-answering style conversation. Conventional sequence-to-sequence neural models have tackled massive tasks for QG, while in CQG, they are liable to lose key information from passage or conversation history, and even tend to produce repetitive words, affecting the answerability of generated questions. To address these issues, we propose a novel multi-source pointer network with coverage mechanism. Our multi-source pointer network has two basic mechanisms: the attention-mixed mechanism jointly calculates the attention distribution of passage and conversation history, which predicts the output probability distribution of words; the copy-generation mechanism decides when to copy a word from the passage or the conversation history and when to generate a word from the predefined vocabulary, which aids accurate copy of information form source text and retains the ability of generating new words. Our coverage mechanism keeps track of what words have been generated, which resolves the repetition in generated questions. Empirical results on CoQA dataset demonstrate that our model outperforms the other state-of-the-art baselines. Moreover, our system effectively reduces the word duplication rates and significantly minimizes the number of generated questions with the repetition.**

I. Introduction

Question Generation (QG) is aiming to automatically generate a fluent and relevant question from the natural langue text [1]. Recently, we have observed that it is receiving increasing interest from both industrial and academic areas. For example, with the development of Question-Answering (QA) systems [2], [3] and conversation systems (virtual assistants, *e.g.* Cortana, Siri, and Google Assistant), QG makes great contributions to the education, *i.e.* the intelligent tutor system, helping teachers set questions according to the given reading comprehension materials for students [4]. Moreover, a system can raise good questions in the conversation will improve user experiences of human-machine interactions [5].

To deeply study the problem of generating conversational questions, a novel dataset, called CoQA, is introduced for building Conversational Question-Answering systems [6]. Soon after, Conversational Question Generation (CQG) is considered as a new scenario in QG [7]. Different from general chatting, under this scenario, a system could ask a series of interconnected questions based on a passage through a question-answering style conversation. Figure 1 provides an example that the conversation is between a questioner and a

Passage: Kendra and Quinton travel to and from school every day. Kendra lives further from the bus stop than Quinton does, stops every morning at Quinton's house to join him to walk to the bus stop. Every afternoon, after school, when walking home from the bus stop they go in for cookies and milk that Quinton's mother has ready and waiting for them. Quinton can't eat cheese or cake so they had the same snack every day. They both work together on their homework and when they are done they play together. Kendra always makes sure to leave in time to get home for dinner. She doesn't want to miss story time which was right before bedtime…

$\mathbf{Q_1}$: Where do Quinton and Kendra travel to and from every day?	$\mathbf{A_1}$: school
$\mathbf{Q_2}$: Does Quinton live further from the bus stop?	$\mathbf{A_2}$: no
$\mathbf{Q_3}$: What do they do every afternoon after school?	$\mathbf{A_3}$: get milk and cookies
$\mathbf{Q_4}$: Does Quinton eat cheese?	$\mathbf{A_4}$: no
$\mathbf{Q_5}$: Do they play before their homework?	$\mathbf{A_5}$: no
$\mathbf{Q_6}$: What does Kendra not want to miss?	$\mathbf{A_6}$: story time
$\mathbf{Q_7}$: When is that?	$\mathbf{A_7}$: right before bedtime
……	……

Fig. 1. An example for conversational question generation on CoQA dataset [6]. Each turn contains a question Q_i and an answer A_i.

answerer, and every question is dependent on the passage and the conversation history. Besides, many questions refer back to the conversation history using coreferences (*e.g.* Q_3, Q_5, Q_7), which is the natural attribute in a human conversation.

In NLP, QG has been mainly coped with a neural approach: using the sequence-to-sequence (or called encoder-decoder) framework to end-to-end train a neural network, *e.g.* ([8], [9], [10]). For example, pointer network [11] has been widely used for machine translation [12], language modeling [13], code generation [14] and text summarization [15], [16], [17], which is a sequence-to-sequence model with attention mechanism solving the problem of generating a sequence whose target dictionary varies depending on the input sequence from single source document. Moreover, a novel sequence-to-sequence architecture, called pointer-generator network [18] is proposed to extend pointer network to decide whether to generate a token from the predefined dictionary or from the input sequence.

However, the goal of our CQG task is to generate interconnected questions in conversational quesiton answering, and a conversational question is dependent on passage and conversation history. Although pointer network works very well in practice, it will lose information from two source documents in CQG, passage and conversation history. So, the first challenge is how to retain key information in the generated questions from both passage and conversation.

Repetition is a common problem for sequence-to-sequence models [19], [20], [21] and is especially pronounced in abstract text summarization [18] task and machine translation [22] task, where generate multi-sentence text. A recent research [23] reveals that such repetition also exists in QG, as shown in Figure 2, repeated occurrence of words 'oxygen' in the input sentence tends to cause repetitions in generated question. Therefore, we consider this problem in CQG and test on the

, H.-N. Dai are with Faculty of Information Technology, Macau University of Science and Technology, Macau. e-mail: (; hndai@ieee.org;).

Paragraph: **Oxygen** storage methods include high pressure **oxygen** tanks, cryogenics and chemical compounds. *For reasons of economy,* ***oxygen*** *is often transported in bulk as a liquid in specially* ***insulated tankers****, since one liter of liquefied* ***oxygen*** *is equivalent to 840 liters of gaseous* ***oxygen*** *at atmospheric pressure and 20 °C (68 °F).* Such tankers are used to refill bulk liquid **oxygen** storage containers, which stand outside hospitals and other institutions with a need for large volumes of pure **oxygen** gas. Liquid **oxygen** is passed through heat exchangers, which convert the cryogenic liquid into gas before it enters the building. **Oxygen** is also stored and shipped in smaller cylinders containing the compressed gas; a form that is useful in certain portable medical applications and oxy-fuel welding and cutting.

Q: by what means is **oxygen** bulk **oxygen** **oxygen** ?

Fig. 2. An example for the repetition in QG (**mark** denotes answers; **mark** denotes sentences that contain answers; **mark** denotes word duplication).

Passage 1: ***The*** ***Food and Agriculture Organization*** *of* ***the*** *United Nations (FAO) is a specialised agency of* ***the*** *United Nations that leads international efforts to defeat hunger.* Serving both developed and developing countries, FAO acts as a neutral forum where all nations meet as equals to negotiate agreements and debate policy. FAO is FAO is also a source of knowledge and information, and helps developing countries in transition modernize and improve agriculture, forestry and fisheries practices, ensuring good nutrition and food security for all . . .

Conversation History (turn=2)

Q$_1$: Where was the first session of the FAO conference held ? **A$_1$**: quebec, canada
Q$_2$: What Chateau was it held in ? **A$_2$**: the chateau frontenac

Seq2Seq:
Q$_3$: what is **the** fao 's name for **the** ? **A$_3$**: the food and agriculture organization

---

Passage 2: ...I know what I want for Chris. I am buying him a football shirt. He is a big Chelsea fan. It's expensive but it's the perfect present. Mum has good ideas about what to buy for Chris's parents. She suggests some typical English foods like tea and marmalade. We find a great shop. It has just what we want. Mum also buys them a photo of London. *That just leaves Chris's sisters. We go to a toy shop and I see some beautiful teddy bears. They are all in different colours and say "Greetings from London." They are perfect. Now I'm ready for my* ***holiday****.*

Conversation History (turn=3)

Q$_1$: What did they purchase for the girls **?** **A$_1$**: teddy bears
Q$_2$: What did they say **?** **A$_2$**: greetings from london
Q$_3$: Were they happy with them **?** **A$_3$**: yes

Seq2Seq:
Q$_4$: what did they want to do now **?** **?** **A$_4$**: his holiday

Fig. 3. Examples for the repetition in CQG (**mark** denotes answers; **mark** denotes sentences that contain answers; **mark** denotes word duplication; **mark** denotes conversation history).

baseline *sequence-to-sequence* model. In order to better show the repetition, we set the minimum length of the generated question to a larger value (8-10 tokens). The result is shown in Figure 3, some repetitive words occur in the generated conversational question, such as the word 'the' and the symbol '?'. And even in Passage2, the pronoun 'they' in generated question Q$_4$ has a reference error (should be 'he'). Hence, how to resolve the repetition issue in CQG is another challenge in our paper.

We put forth a novel encoder-decoder model named *Multi-Source Pointer Network* (MS-PNet) that incorporates two encoders, passage encoder and conversation encoder, and one decoder with attention-mixed mechanism, copy-generation mechanism, and coverage mechanism to generate conversational questions.

The main contributions of our paper can be summarized as following:

- We propose an Attention-Mixed Mechanism that combining the Bahdanau attention mechanism [24] with the hierarchical attention mechanism [25] to decide to focus on passage hidden states or conversation hidden states and uses an additional attention layer to update the output hidden states while calculating the attention distribution.
- We introduce a Copy-Generation Mechanism that helps copying words from passage or conversation via pointing [11] and generating words from vocabulary, which improves accuracy and handling of out-of-vocabulary (OOV) words, while retaining the ability of generating novel words.
- We adopt a variant Coverage Mechanism [18] from Abstractive Text Summarization to track and control coverages of our two source documents, passage and conversation, and define coverage loss via JS divergence to penalize the repetition. The final experimental results show that our coverage mechanism is significantly effective for eliminating the repetition.

The remainder of this paper is organized as follows. In Section II, we describe the overview of our framework and the main proposed approaches in details. Then the experiments evaluation and results are discussed are discussed in Section III. Finally, we conclude our work and introduce the future research directions in Section IV.

II. Our Model

As shown in Figure 4, our model consists of two encoders, (1) passage encoder and conversation encoder, and one decoder with (2) attention-mixed mechanism, (3) copy-generation mechanism, and (4) coverage mechanism.

A. Task Definition

We use P, $C_{j-1} = \{(Q_1, A_1), \ldots, (Q_{j-1}, A_{j-1})\}$, A_j to represent passage, conversation history, and current answer respectively, and our task is to generate next turn question $\widetilde{Q}_j$:

$$\widetilde{Q}_j = \arg\max_{Q_j} \mathrm{Prob}(Q_j \mid P, C_{j-1}, A_j), \qquad (1)$$

where passage is comprised of sequence of words, answer phrases are text fragments in passage, and words in question are generated either from passage or from conversation history. Note that in our task setting, answers are given before generating questions, which is known as answer awareness QG problem [9], [23], [26].

B. Passage and Conversation Encoding

For CQG, there are two source documents: passage and conversation history. So, our multi-source pointer network has two encoders, passage encoder and conversation encoder, jointly encodes information from two sources.

Passage Encoder. Our passage encoder is a bidirectional-LSTM. The tokens of passage are fed one-by-one into passage encoder, producing a sequence of encoder hidden states

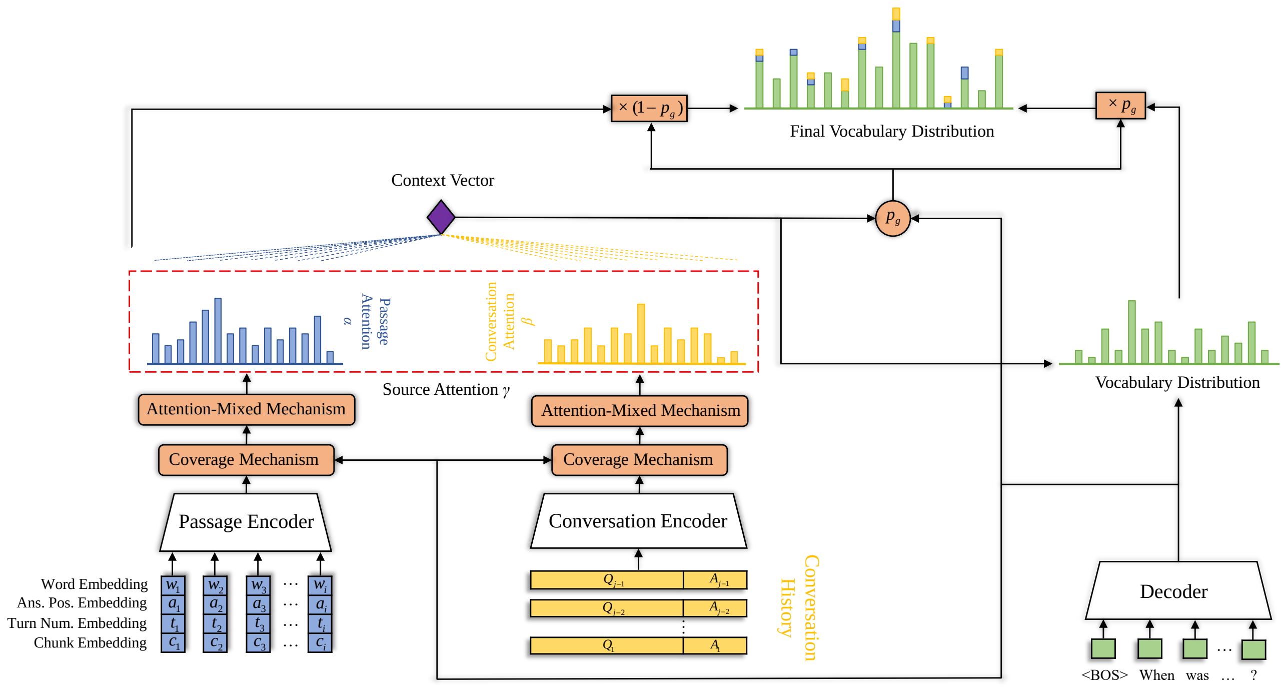

Fig. 4. Multi-Source Pointer Network with Coverage Mechanism. At each decoding timestep, a generation probability $p_g \in [0, 1]$ is calculated, which weights the probability of generating words from the vocabulary, versus copying words from the source texts (passage and conversation). The vocabulary distribution and the two attention distributions are weighted and summed to derive the final distribution, helping us make final prediction. *(Best viewed in color)*

$(h_1^p, \ldots, h_m^p)$ as the passage representation, where m is the sequence length.

a) Input Embeddings: We take the concatenation of word embedding w and answer position embedding a as input $x_i = [w_i; \mathrm{a}_i]$. Besides, to enable the conversation to have smooth transitions between turns, we model the conversation flow via two extra embeddings, which is similar to the techniques in [7]. The first turn number embedding is a lookup table $[t_1, \ldots, t_n]$, where n is the maximum turn. Then, we split the passage into l unified chunks and get its corresponding chunk embedding $[c_1, \ldots, c_l]$. Therefore, the final input to our passage encoder is $x_i = [w_i; \mathrm{a}_i; t_i; c_i]$, which concatenates the word embedding, the answer position embedding, the turn number embedding and the chunk embedding.

b) Gated Self-attention Mechanism: We also introduce a gated self-attention mechanism [23] to aggregate more answer-related information for question generation. We match the encoded passage representation h_i^p with rich-feature enhanced passage representation $H^p = [h_1^p; \ldots; h_m^p]$ to obtain self-matching representation s_i^p:

$$\mathrm{a}_i^p = \mathrm{softmax}(H^{p^\top} W^p h_i^p) \tag{2}$$

$$s_i^p = H^p \mathrm{a}_i^p \tag{3}$$

Then combining it with the original passage representation h_i^p as the new self-matching representation f_i^p, and compute a learnable gate vector g_t^p. Finally, we can derive the final passage representation $\tilde{h}_i^p$:

$$f_i^p = \tanh(W^f [h_i^p; s_i^p]) \tag{4}$$

$$g_t^p = \mathrm{sigmoid}(W^g [h_i^p; s_i^p]) \tag{5}$$

$$\tilde{h}_i^p = g_t^p \odot f_i^p + (1 - g_t^p) \odot h_i^p \tag{6}$$

where W^p, W^f, W^g are learnable parameters, $\odot$ is the element-wise multiplication. Please refer to Zhao *et al.* [23] for more details on gated self-attention mechanism.

Conversation Encoder. Since conversation history C_{j-1} has several turns and each turn has related question-answer (QA) pairs $\{(Q_1, A_1), \ldots, (Q_{j-1}, A_{j-1})\}$, we use a hierarchical structure to explore the contextualized representation of conversation history during encoding. We first adopt a bi-LSTM to learn the contextual representation of question-answer pairs $(h_{j-k,1}^w, \ldots, h_{j-k,m}^w)$ on token level, where $j-k$ is the turn number and $k \in [1, j)$. Then, we employ a bi-LSTM to obtain the contextual representation of dependency across different turns of question-answer pairs $(h_1^c, \ldots, h_{j-1}^c)$, where subscript denotes the turn number.

C. Decoding with Multiple Mechanisms

Our decoder is a double-layer unidirectional LSTM. At each decoding timestep t, it receives the word embedding w and previous hidden state h^d_{t-1} to generate current decoder state h^d_t:

$$h^d_t = \mathrm{LSTM}(w, h^d_{t-1}) \tag{7}$$

Attention-Mixed Mechanism. Some existing attention mechanisms [24], [27], [25] have been widely used to improve sequence-to-sequence models' performance for many applications. In our task, since a conversational question depends on both passage and conversation history, we propose an attention-mixed mechanism to help our decoder decide where to look, passage or conversation, to produce next word.

During the encoding stage, our passage encoder has produced a sequence of encoder hidden states $\tilde{h}^p_i$, and conversation encoder has produced a sequence of encoder hidden states, $h^w_{j-k,i}$ and h^w_{j-k}, respectively on token level and context level. Therefore, at each decoding timestep, we first compute the raw attention scores of passage e^p_i and the raw attention scores of conversation, $e^w_{j-k,i}$ and e^c_{j-k} as in [24]:

$$e^p_i = v^T_p \tanh(W_p \tilde{h}^p_i + W_{d_1} h^d_t + b_p) \tag{8}$$

$$e^w_{j-k,i} = v^T_w \tanh(W_w h^w_{j-k,i} + W_{d_2} h^d_t + b_w) \tag{9}$$

$$e^c_{j-k} = v^T_c \tanh(W_c h^c_{j-k} + W_{d_3} h^d_t + b_c) \tag{10}$$

where v_p, v_w, v_c, W_p, W_w, W_c, W_{d_1}, W_{d_2}, W_{d_3}, b_p, b_w, b_c are learnable parameters. Then we use the hierarchical attention mechanism for raw conversation attention to calculate the total raw attention scores of source-side, and get the passage attention α_i and the conversation history $\beta_{j-k,i}$:

$$e_{\text{total}} = \Sigma_i e^p_i + \Sigma_{k,i} e^w_{j-k,i} * e^c_{j-k} \tag{11}$$

$$\alpha_i = \frac{e^p_i}{e_{\text{total}}} \tag{12}$$

$$\beta_{j-k,i} = \frac{e^w_{j-k,i} * e^c_{j-k}}{e_{\text{total}}} \tag{13}$$

In this way, we obtain the passage attention distribution $(\alpha_1, \ldots, \alpha_m)$, the conversation attention distribution $(\beta_{1,1}, \ldots, \beta_{1,m}; \ldots; \beta_{j-1,1}, \ldots, \beta_{j-1,m})$, and the source-side attention distribution $\gamma^t = [\alpha^t; \beta^t]$ at each decoding timestep t. Next, we conduct a weighted sum of two encoder hidden states, known as the context vector c_t:

$$c_{\text{t}} = \Sigma_i \alpha^t_i \tilde{h}^p_i + \Sigma_{k,i} \beta^t_{j-k,i} h^w_{j-k,i} \tag{14}$$

An additional attentional layer is applied to combine the concatenation of decoder state h^d_t and context vector c_t to produce new decoder hidden state $\tilde{h}^d_t$, and fed through the softmax layer to derive the vocabulary distribution P_v:

$$\tilde{h}^d_t = \tanh(W_a[h^d_t, c_t]) \tag{15}$$

$$P_v = \mathrm{softmax}(W_v \tilde{h}^d_t + b_v) \tag{16}$$

where W_v, b_v are learnable parameters. P_v is a probability distribution of the words in the pre-defined vocabulary, and we can use it to predict words:

$$P(w) = P_v(w) \tag{17}$$

Copy-Generation Mechanism. Copy mechanism [28] and pointer network [18] allow both copying words from source input and generating words from a pre-defined vocabulary (corpus) during decoding. In order to let generated words are not restricted from the fixed vocabulary, we extend final word vocabulary to a union of the pre-defined vocabulary and all words from the source-side documents (passage and conversation history).

Similar to [18], given the decoder input x_t, the context vector c_t and the decoder state $\tilde{h}^d_t$, our copy-generation mechanism computes the generation probability p_g at the timestep t:

$$p_{\text{g}} = \mathrm{sigmod}(w^T_c c_t + w^T_d \tilde{h}^d_t + w^T_x x_t + b_g) \tag{18}$$

where w_c, w_d, w_x, b_g are learnable weights and $p_g \in [0, 1]$. Then, p_g can be used as a flexible switch to choose to generate a word from the fixed vocabulary via the vocabulary distribution P_v, or to copy a word from source documents via the source-side attention distribution γ^t.

$$\widetilde{P}(w) = p_{\text{g}} P_{\text{v}}(w) + (1 - p_{\text{g}}) \sum_{i:w_i=w} \gamma^t_i \tag{19}$$

Note that if w appears in the source-side documents, then $P_v(w)$ is zero; similarly, if w is not an out-of-vocabulary (OOV), $\sum_{i:w_i=w} \gamma^t_i$ is zero.

Coverage Mechanism. Repetition is a common problem which has already been observed in some sequence-to-sequence models [19], [20], [18], [23]. In this paper, we also consider the repetition on conversational question generation and build the coverage model which is similar to Tu *et al.* [19] to solve the problem of word duplication.

Our coverage model maintains two coverage vectors $(c_\alpha)^t$ and $(c_\beta)^t$, that respectively record historical passage attention weight distributions and conversation attention weight distributions calculated in all previous decoder timesteps:

$$(c_\alpha)^t = \mu_\alpha \cdot (c_\alpha)^{t-1} + (1 - \mu_\alpha) \cdot \exp(-\eta_1 \alpha^{t-1}) \tag{20}$$

$$(c_\beta)^t = \mu_\beta \cdot (c_\beta)^{t-1} + (1 - \mu_\beta) \cdot \exp(-\eta_2 \beta^{t-1}) \tag{21}$$

where μ_α, μ_β, η_1, η_2 are hyperparameters. Note that c^0 is a zero vector, because on the first timestep, none of the source document has been covered. The purpose of coverage vectors is recording which source word has been paid attention to in the source text, and tell our model to avoid that continuous attention attends to the same location to generate repetitive word. Therefore, our two coverage vectors are used as extra inputs to update our attention-mixed mechanism, changing equation (8), (9) to:

$$e_i^p = v_p^T \tanh(W_p \tilde{h}_i^p + W_{d_1} \tilde{h}_t^d + w_{c_1} (c_\alpha)_i^t + b_p) \tag{22}$$

$$e_{j-k,i}^w = v_w^T \tanh(W_w h_{j-k,i}^w + W_{d_2} \tilde{h}_t^d + w_{c_2} (c_\beta)_{j-k,i}^t + b_w) \tag{23}$$

where w_{c_1}, w_{c_2} are learnable weights. Since in the process of generating questions, the current larger attention weight makes greater contributions to predicting next word. In this way, when our model calculates the attention distribution, our coverage mechanism will attenuate the attention weights for the next same locations according to the historical information, which allowing our model to pay more attention to different source words informed by the previous decisions.

D. Joint Training.

Conversation Flow Loss.

During the encoding stage, we build a conversation flow model on passage. To better help our model generate conversational questions in each turn, we design a loss to focus which sentences around answers are of high informativity.

Firstly, we label two kinds of sentences in the passage. If a sentence is informative to the current question, called Current Informativity Sentence (CIS). Otherwise, called History Informativity Sentence (HIS). Then we introduce a conversation flow loss to enable our model to locate the correct informativity sentences:

$$\mathcal{L}_{\text{flow}} = -\lambda_3 \log \frac{\sum_{i,w_i \in \text{CIS}} \alpha_i}{\sum_i \alpha_i} + \lambda_4 \frac{\sum_{i:w_i \in \text{HIS}} \alpha_i}{\sum_i \alpha_i} \tag{24}$$

where λ_1, λ_2 are hyperparameters, and w_i is a token with CIS/HIS label in the sentence if $w_i \in$ CIS/HIS.

Negative Log-Likelihood Loss with Coverage Loss.

In the decoding stage, we compute the final probability distribution $\widetilde{P}(w)$ over the extended vocabulary. At timestep t, the negative log likelihood loss for predicted word w_t' is:

$$\text{loss}_t = -\log \widetilde{P}(w_t') \tag{25}$$

Since we also introduce coverage mechanism, it's necessary to additionally define two coverage losses to penalize the overlap. We use JS divergence to measure the distance between coverage vectors and attention distributions because they both are probability distribution:

$$\text{covloss}_t^\alpha = -D_{JS}(\alpha^t \| \operatorname{softmax}((c_\alpha)^t)) \tag{26}$$

$$\text{covloss}_t^\beta = -D_{JS}(\beta^t \| \operatorname{softmax}((c_\beta)^t)) \tag{27}$$

We take the negative value of this distance as the loss function, so as to increase the difference between two distributions, and reduce the possibility of attending to the repeated locations. Then, we add two coverage losses to NLL loss to yield a composite loss reweighted by hyperparameters λ_5, λ_6:

$$\mathcal{L}_{\text{nll+cov}} = -\log \widetilde{P}(w_t') + \lambda_3 \text{covloss}_\alpha^t + \lambda_4 \text{covloss}_\beta^t \tag{28}$$

Coreference Alignment Loss.

Generating coreference words, such as *he*, *she*, or *it* in CoQA [6], is a distinctive characteristic in CQG task. So, we adopt a coreference alignment model proposed by Gao *et al.* [7] to achieve this goal.

We first use coreference resolution system [29] to find coreference words $(w_1^c, \ldots, w_i^c)$ between the conversation history C_{j-1} and question Q_j in the preprocessing stage. Then, we introduce a loss function built on the conversation attention of coreference words β_i^c and their word probability p_{coref} ($p_{\text{coref}} \in P_v$) during the decoding stage:

$$\mathcal{L}_{\text{coref}} = -(\lambda_5 \log \frac{\Sigma_i \beta_i^c}{\sum_{k,i} \beta_{j-k,i}} + \lambda_6 \log p_{\text{coref}}) * s_c \tag{29}$$

where λ_1, λ_2 are hyperparameters, s_c is the confidence score. Considering the all aforementioned losses, we derive a final joint loss function for our model during training:

$$\mathcal{L} = \mathcal{L}_{\text{flow}} + \mathcal{L}_{\text{nll+cov}} + \mathcal{L}_{\text{coref}} \tag{30}$$

III. Experiment

We conduct our experiments on a large-scale dataset CoQA [6] for conversational question generation. They collet 127k questions with answers, obtained from 8k conversations about text passages from seven diverse domains. We first extract passage, question, conversation history (QA pairs) from CoQA dataset, and then filter out some QA pairs such as 'yes' or 'no' QA pairs. Finally, we shuffle the dataset and split it a into training set (80%, 66298 samples), a development dataset (10%, 8409 samples) and a testing set (10%, 8360 samples). The details on our data preprocessing are similar to Gao *et al.* [7] and the processed dataset is supported at https://github.com/Yifan-Gao/conversational-QG.

A. Experiment Settings

We use double-layers bi-directional LSTM for decoding and passage and token-level QA pairs encoding, but use single-layer un-directional LSTM for sentence-level QA pairs encoding. For word embedding, we use pre-trained GloVe word vectors with 300 dimensions [30]. Both encoder and decoder share the same source-word vocabulary of 59523 words. For optimization, we use AdaGrad. Learning rate is initially set to 0.1 and decay rate is set to 0.5. The batch size for training is 32 and for validating is 16. After training, we choose a model with 20000 training steps as the final model. During decoding for generation, we use beam search with the beam size 5 and set the maximum length for output sentence as 15. To guarantee the target questions have enough conversation history information to generate, we choose the number of history turns as $n = 3$. Moreover, when we evaluate the effectiveness of coverage mechanism on our model, we set the minimum length to 10. All hyperparameters and models are selected on the development set and the results are reported on the test set.

TABLE I. Main results of QG baselines and our models on test dataset (**bold** denotes the best scores).

	BLEU_1	BLEU_2	BLEU_3	ROUGE_L
PGNet	28.84	13.74	8.16	39.18
NQG	35.56	21.14	14.84	45.58
MS-PNet	36.40	21.94	15.43	46.31
MS-PNet+Cov	35.80	21.42	15.08	46.00
MS-PNet+Cov(JS)	**36.43**	**21.98**	**15.56**	**46.39**

TABLE II. Main results of QG baselines and our models on coreference dataset (**bold** denotes the best scores).

	BLEU_1	BLEU_2	BLEU_3	ROUGE_L
PGNet	27.66	13.82	8.96	38.40
NQG	34.75	21.52	15.96	45.04
MS-PNet	36.71	23.30	17.46	46.71
MS-PNet+Cov	36.36	23.16	17.53	46.82
MS-PNet+Cov(JS)	**37.56**	**24.10**	**18.17**	**47.58**

B. Baselines and Ablations

Our experiments aim to investigate if our method can improve the performance of generating conversational questions and meanwhile reducing the repetition in generated questions. So, we introduce several QG baseline models as follows:

PGNet is a sequence-to-sequence attentional model [18] using a hybrid pointer network that can copy words from the source text and generate words from the pre-defined vocabulary. We change their input by concatenating the passage P, the conversation history C_{j-1} and the current answer A_j.

NQG [31] is an attention-based sequence learning model, which is similar to **PGNet**, but during encoding it concatenates the answer features, coreference position features and the word embeddings as the input.

MS-PNet is our base model **M**ulti-**S**ource **P**ointer **Net**work (with attention-mixed mechanism and copy-generation mechanism).

MS-PNet+Cov is our base model with **Cov**erage mechanism proposed by See *et al.* [18] in abstractive summarization.

MS-PNet+Cov(JS) is the base model **MS-PNet** with our novel **Cov**erage mechanism and **JS** divergence loss.

C. Experiment Evaluation Metrics

We use n-gram similarity metrics BLEU(1-3) [32], and ROUGE-L [33] to evaluate the performance of generating on our model and basslines.

D. Experiment Results and Analyses

As we discussed that generating questions with the pronouns is a basic characteristic of conversational question generation. So, we also evaluate models on coreference dataset (a subset of the test dataset), using the same evaluation metrics. Each sample in the coreference dataset requires a pronoun resolution between the conversation history and the current question (*e.g.* Q_3, Q_5, Q_7). Table I shows the main results of various models on our dataset and Table II shows the results on coreference dataset. We can observe that:

- **NQG** has larger improvements than **PGNet**, which shows that the answer position embedding and coreference position embedding make great contributions for conversational question generation.
- **MS-PNet** is our base model and outperforms two baselines (**PGNet** and **NQG**) on both test dataset and coreference dataset, which reveals that our attention-mixed mechanism can establish the relationship between passage attention and conversation attention, and copy-generation can help keep key information from passage and conversation history. We will discuss these two mechanism later.
- **MS-PNet+Cov** almost has a declining performance on test dataset and coreference dataset compared with our base model, since adding the coverage mechanism [18] from abstractive summarization. However, the main purpose of coverage mechanism is aming to reducing the repetition, we will analyze the effectiveness of their coverage model and compared with our approach later.
- **MS-PNet+Cov(JS)** is significantly better than the all aforementioned models although we adopt our coverage mechanism to resolve the repetition issue at the same time.

Attention-Mixed Mechanism Analysis

Our attention-mixed mechanism combines two attention mechanisms: the Bahdanau attention mechanism [24] and the hierarchical attention mechanism [25]. Main idea is using the 'concat' alignment function in [24] to calculate the scores of attention for passage and conversation history. However, conversation history has a hierarchical encoding, we use the hierarchical attention on conversation history to model the dependency across different turns in conversations. And finally, we employ an additional layer, which is the similar technique in [27], to update the output of hidden states. The differences from the standard Bahdanau attention mechanism [24] are: (1) we use the current hidden state (rather than the previous state) at each decoding timestep to calculate attention. (2) we combine with hierarchical attention to calculate the attention together (3) to improve the integrity and relevance

Example 1: ...Cooking came easy to Kenny, and he enjoyed adding new ingredients into common dishes. "It started with a passion and I wanted to know more," he said. He began to watch the Food Network and read chef blogs. *Last summer, Kenny put his skills to the test by* ***working with his grandmother*** *to cater his uncle's wedding.* While she cooked traditional dishes, Kenny wanted to add new to the expected flavors. Now Kenny spends his weekends catering his own events:..

Conversation History (turn=3)

Q_1: Was did he cook first?	A_1: shrimp and broccoli
Q_2: Where did he eat it?	A_2: school
Q_3: What TV station did he watch?	A_3: Food Network

MS-PNet:
Q_4: what did he **do** last summer ?

Examle 2: ...It turns slightly several times, and comes to rest at the rear of her brain. And in the process, the bullet misses all the vital areas of the brain. In many ways, it almost misses the brain itself," he said. ***"In Christianity*** *we* ***call it prevenient grace****: God working ahead of time for a particular event in the future.* It's just like the God I follow to plan the route of a bullet through a brain long before Batman ever rises. Twenty-two years before," Strait wrote.

Conversation History (turn=3)

Q_1: How old was she?	A_1: 22
Q_2: What was a miracle?	A_2: brain abnormality
Q_3: Was her pastor involved?	A_3: No

MS-PNet
Q_4: what did he **call it in christianity** ?

Fig. 5. An example for copy-generation network (**mark** denotes answers; **mark** denotes sentences that contain answers; **mark** denotes novel words; mark denotes conversation history; **mark** denotes copied words).

Passage 1: Bob had a son that was named John. John owned the only red **dog**. *There was a blue* ***dog*** *named* ***Jim***. The red **dog** was named Woof. John's mother and Bob's wife was named Gail. They all lived in a tree. There was a family down the street. They were a couple named Frank and Francine. Bob hated his neighbors. One day Bob went to Frank's house and threw toilet paper all over his trees. Frank got back at Bob by stealing his son's **dog**. This is how the long fight between Frank and Bob began. It lasted for ten years until Bob passed away from a bad heart attack...

Target: what is the name of the blue dog ?
MS-PNet: what is the blue **dog** name of **dog dog** ?
MS-PNet+Cov(JS): what is the name of blue dog ?

Passage 2: ..."Most of **the** customers were not aware there is an ATM in Somalia," Kisaulu explained. "Local residents, most of them don't have a debit credit or a credit card to use **the** ATM." **The** U.S. dollar is "widely accepted" in Mogadishu and is used in local trading, according to **the** African Development Bank. ***The*** *country's official currency,* ***the Somali shilling****, has seen a sustained decline in value following decades of conflict in* ***the*** *war-torn nation.* Said Moallim Abukar, marketing manager of Salaam Somali Bank, which operates **the** ATM, said that other currencies would be available from **the** machine "in **the** near future." But Michael Walls...

Target: what is Somalia's official coinage ?
MS-PNet: what is **the** official currency in value of **the** ?
MS-PNet+Cov(JS): what is the country 's official currency called ?

Fig. 6. Examples for **MS-PNet** with our coverage mechanism vs. basic **MS-PNet** (**mark** denotes answers; **mark** denotes sentences that contain answers; **mark** denotes word duplication).

of features, we use an additional layer to take context vector and the hidden state together to update the output in decoder.

As results in Table I and Table II, our attention mechanism performs better than single attention mechanism used in baseline models. It helps our **MS-PNet** model decide to focus on passage or conversation history in CQG.

Copy-Generation Mechanism Analysis

The pointer mechanism [18] works well in summarization task, such as OOV words are handled easily and factual details are almost copied correctly. In our CQG task, two examples are depicted in Figure 5 that our model can generate a novel word 'do' which is not copied from passage or conversation history in Example1 and also can copy words, such as 'call', 'it', 'in', and 'christianity' from passage in Example2.

Therefore, we can prove that our copy-generation mechanism can help our model **MS-PNet** well copy words from source text, while retaining the ability of producing new words in CQG task, which enriches the vocabulary of generated questions.

Coverage Mechanism Analysis

Our coverage mechanism is designed to resolve the word repetition issue bought by our copy-generation mechanism. As shown in Figure 6, when we set the minimum length of generated question to a large value, our basic **MS-PNet** model will be more likely to generate some repetitive words compared with coverage model. It really affects the quality of generated questions. Coincidentally, such repetition issue has been also observed in abstractive summarization task, e.g [18] and they proposed a coverage model and related coverage loss to resolve the repetition. So, we also implemented and tested their approach in our CQG task.

Actually in QG task, Zhao *et al.* [23] proposed a maxout pointer mechanism and demonstrated that it really can reduce word duplication in generated question, which provides an approach for us to evaluate our coverage model in CQG task. Therefore, we first select out every generated question which has word duplication and label "Repetitive Question". Then we compute the rate of the word duplication via counting the number of words which appear more than once, taking a ratio of them over the total word counts. Besides, we count the number of such questions and compute the average ratio of the total number of generated questions. In order to better demonstrate the effectiveness of our coverage mechanism and the answerability of the generated questions, we set the minimum length of generated questions between 8 and 10. And then, we evaluate general questions and coreference questions generated from test dataset and coreference dataset, respectively.

As shown in Figure 7, compared with our base model **MS-PNet** and **MS-PNet** with coverage mechanism proposed by See *et al.* [18], our **MS-PNet+Cov(JS)** can slightly reduce the ratio of word duplication. Moreover, in Figure 8, we can observe that our coverage mechanism significantly reduces the number of generated questions that have repetitive words, whatever on our test dataset or coreference dataset. Specifically, the effectiveness of coverage model [18] in CQG task is not better than ours, even though the duplication ratio and the number dropped as expected. Besides, our base model without coverage mechanism generates about twice as many conversational questions with word duplication as our coverage model. So, our coverage mechanism really makes contributions to reducing the repetition in conversational question generation.

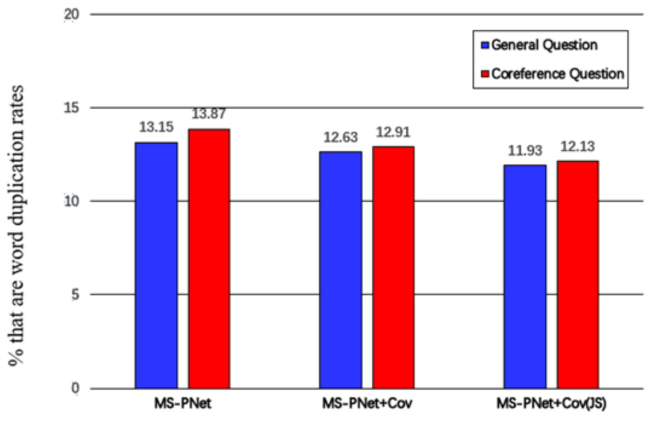

Fig. 7. Word duplication rates of "Repetitive Question".

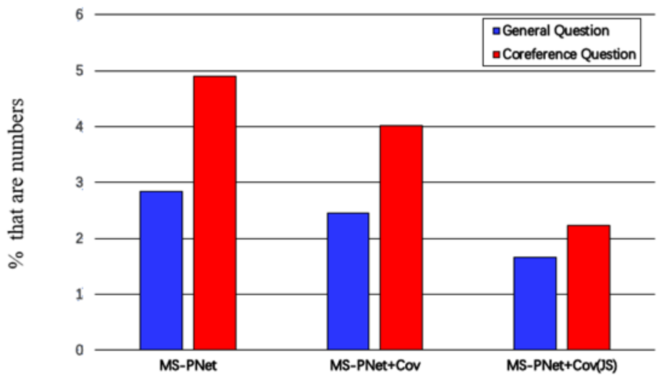

Fig. 8. The number of "Repetitive Question".

IV. Conclusion

In this paper, we study conversational question generation (CQG) and the problem of repetition in CQG, which has never been investigated before. We propose a novel multi-source pointer network with attention-mixed mechanism and copy-generation mechanism, and also introduce our coverage mechanism to solve the repetition issue. When we use attention-mixed mechanism to calculate the attention of the source documents (passage and conversation history), our coverage mechanism will maintain two coverage vectors, avoiding the continuous attention attends to the same location to predict the repetitive word. Moreover, our copy-generation mechanism allows both copying words and generating news words to generate conversational questions. Experiments show that although the coverage mechanism [18] makes some sense on the word duplication, it degrades the performance in CQG task. Our model really has a great performance in generating conversational questions and is also effective in reducing the word repetition rates and the number of questions generated with the repetition.

In the future, we intend to study (1) how to generate conversational questions with the short or long length under the guarantee of fluency and answerability; (2) how to identify the characteristic (*e.g.* gender) of antecedent words and use the correct pronouns to refer back; (3) how to enrich CQG scenario settings, such as questions can be raised by either part, to improve user experience and human-machine interaction; (4) what is more interesting is how to apply CQG task to the examination. For example, the system can generate difficulty-controlled questions according to candidates' previous answers, which better serves the education industry.

References

[1] M. Heilman and N. A. Smith, "Good question! statistical ranking for question generation," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 609–617.

[2] N. Duan, D. Tang, P. Chen, and M. Zhou, "Question generation for question answering," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 866–874.

[3] H.-Y. Shum, X.-d. He, and D. Li, "From eliza to xiaoice: challenges and opportunities with social chatbots," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 10–26, 2018.

[4] X. Du, J. Shao, and C. Cardie, "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)*, 2017, pp. 1342–1352.

[5] Y. Wang, C. Liu, M. Huang, and L. Nie, "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)*, 2018, pp. 2193–2203.

[6] S. Reddy, D. Chen, and C. D. Manning, "Coqa: A conversational question answering challenge," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 249–266, 2019.

[7] Y. Gao, P. Li, I. King, and M. R. Lyu, "Interconnected question generation with coreference alignment and conversation flow modeling," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 4853–4862.

[8] X. Yuan, T. Wang, C. Gulcehre, A. Sordoni, P. Bachman, S. Zhang, S. Subramanian, and A. Trischler, "Machine comprehension by text-to-text neural question generation," in *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 2017, pp. 15–25.

[9] Q. Zhou, N. Yang, F. Wei, C. Tan, H. Bao, and M. Zhou, "Neural question generation from text: A preliminary study."

[10] L. Song, Z. Wang, and W. Hamza, "A unified query-based generative model for question generation and question answering," *arXiv preprint arXiv:1709.01058*, 2017.

[11] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," *arXiv preprint arXiv:1506.03134*, 2015.

[12] C. Gulcehre, S. Ahn, R. Nallapati, B. Zhou, and Y. Bengio, "Pointing the unknown words," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 140–149.

[13] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," *arXiv preprint arXiv:1609.07843*, 2016.

[14] W. Ling, P. Blunsom, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, F. Wang, and A. Senior, "Latent predictor networks for code generation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 599–609.

[15] Y. Miao and P. Blunsom, "Language as a latent variable: Discrete generative models for sentence compression," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 319–328.

[16] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gùlçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence rnns and beyond," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, 2016, pp. 280–290.

[17] W. Zeng, W. Luo, S. Fidler, and R. Urtasun, "Efficient summarization with read-again and copy mechanism," *arXiv preprint arXiv:1611.03382*, 2016.

[18] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017, pp. 1073–1083.

[19] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 76–85.

[20] B. Sankaran, H. Mi, Y. Al-Onaizan, and A. Ittycheriah, "Temporal attention model for neural machine translation," *arXiv preprint arXiv:1608.02927*, 2016.

[21] J. Suzuki and M. Nagata, "Cutting-off redundant repeating generations for neural abstractive summarization," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017, pp. 291–297.

[22] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[23] Y. Zhao, X. Ni, Y. Ding, and Q. Ke, "Paragraph-level neural question generation with maxout pointer and gated self-attention networks," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3901–3910.

[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[25] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.

[26] X. Sun, J. Liu, Y. Lyu, W. He, Y. Ma, and S. Wang, "Answer-focused and position-aware neural question generation," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3930–3939.

[27] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.

[28] J. Gu, Z. Lu, H. Li, and V. O. Li, "Incorporating copying mechanism in sequence-to-sequence learning," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1631–1640.

[29] K. Clark and C. D. Manning, "Deep reinforcement learning for mention-ranking coreference models," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2256–2262.

[30] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[31] X. Du and C. Cardie, "Harvesting paragraph-level question-answer pairs from wikipedia," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 1907–1917.

[32] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[33] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.