Keep and Select: Improving Hierarchical Context Modeling for Multi-Turn Response Generation

Yanxiang Ling, Fei Cai, Jun Liu, *Senior Member, IEEE*, Honghui Chen, and Maarten de Rijke

***Abstract*— Hierarchical context modeling plays an important role in the response generation for multi-turn conversational systems. Previous methods mainly model context as multiple independent utterances and rely on attention mechanisms to obtain the context representation. They tend to ignore the explicit responds-to relationships between adjacent utterances and the special role that the user's latest utterance (the query) plays in determining the success of a conversation. To deal with this, we propose a multi-turn response generation model named KS-CQ, which contains two crucial components, the Keep and the Select modules, to produce a neighbor-aware context representation and a context-enriched query representation. The Keep module recodes each utterance of context by attentively introducing semantics from its prior and posterior neighboring utterances. The Select module treats the context as background information and selectively uses it to enrich the query representing process. Extensive experiments on two benchmark multi-turn conversation datasets demonstrate the effectiveness of our proposal compared with the state-of-the-art baselines in terms of both automatic and human evaluations.**

***Index Terms*— Hierarchical context modeling, multi-turn conversational system, neural generative model, response generation.**

I. Introduction

IN RECENT years, neural response generation has attracted considerable interests in the field of open-domain conversational systems [1], due to the availability of large-scale corpora and recent progress in deep learning technologies [2], [3]. Compared with single-turn scenario, multi-turn conversations [4]–[12] are more extensive in daily life and come with stricter requirements for contextual coherence. In a multi-turn scenario, response generation should not only depend on the user's latest utterance (the query) but also be consistent with conversational history (the context). Thus, how to model context and query is key to multi-turn response generation.

There are mainly two kinds of context modeling methods: non-hierarchical and hierarchical. The non-hierarchical methods usually concatenate contextual utterances into one sentence using their chronological order [4]–[7] or rewrite them as a new informative sentence [13], [14], and then feed the sentence into a vanilla sequence-to-sequence framework [15] to generate a response. These methods essentially follow the single-turn framework, which may neglect the dynamic topic flow across utterances [8], [16]. Thus, to further investigate the semantic structure of contextual utterances, hierarchical context modeling methods [8]–[12] have been proposed, which model the context at both utterance and discourse levels. A previous work has incorporated memory networks [17], latent variable models [18] and variational autoencoders [19], [20] into the hierarchical framework. Compared with the non-hierarchical methods, the hierarchical methods have shown better performance on capturing conversational dynamics and have achieved improvements in multi-turn response generation [3], [9], [12].

One challenge of multi-turn context modeling is to obtain the semantic representation of context [1]. The current hierarchical models usually regard context as multiple independent utterances and encode them separately. They ignore the fact that a multi-turn conversation is produced in a coherent process, where utterances are semantically related and mutually complementary. As illustrated in Table I, adjacent utterances in a conversation, e.g., (Utterance 1, Utterance 2), (Utterance 2, Utterance 3), have explicit responds-to relationships. When encoding utterances separately without consideration of their inner relationships, hierarchical models may fail to capture the discourse coherence within context and eventually produce non-ideal responses, as shown in Table I. In addition, conversational utterances tend to be colloquial with omissions, e.g., in Utterance 3, "Yes. What did you think of it?" The word "it" refers to " the new James Bond movie." We hypothesize that encoding utterances separately will produce nonspecific utterance representations and lead to the generation of uninformative or irrelevant responses.

Another challenge of multi-turn context modeling is to detect relevant context for the ongoing response generation [12], [21], [22], since the contribution of different context to response generation is likely to change as the conversation progresses, especially for conversations with many turns. Previous hierarchical models mostly depend on response-context [8], [11], [12] and context-context attention mechanisms [10], [12] to detect relevant context. Unfortunately, this is not a guarantee for generating relevant responses, as shown in Table I. The query (such as Utterance 4 in Table I) is a promising source to capture the focus of an ongoing conversation, since the response is fundamentally generated

Manuscript received August 30, 2020; revised August 16, 2021; accepted September 8, 2021. This work was supported by the National Natural Science Foundation of China under Grant 61702526. *(Corresponding author: Fei Cai.)*

Yanxiang Ling is with the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China, and also with the College of Information Communication, National University of Defense Technology, Xi'an 710102, China (e-mail: lingyanxiang@nudt.edu.cn).

Fei Cai and Honghui Chen are with the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China (e-mail: caifei08@nudt.edu.cn; chenhonghui@nudt.edu.cn).

Jun Liu is with the Department of Computer Science, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: liukeen@xjtu.edu.cn).

Maarten de Rijke is with the Informatics Institute, University of Amsterdam, 1012 XM Amsterdam, The Netherlands (e-mail: m.derijke@uva.nl).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3112700.

Digital Object Identifier 10.1109/TNNLS.2021.3112700

TABLE I

EXAMPLE OF A MULTI-TURN CONVERSATION AND RESPONSES GENERATED BY VARIOUS MODELS. REFERENCE IS THE REAL RESPONSE PROVIDED IN THE DATASET. HRED, WSEQ, SD-HRED, HRAN, AND RECOSA ARE STRONG HIERARCHICAL MODELS, WHERE RECOSA IS THE STATE-OF-THE-ART WORK. KS-CQ IS THE PROPOSED METHOD

Conversation	
Utterance 1:	Have you seen the new James Bond movie?
Utterance 2:	Oh, yes. You?
Utterance 3:	Yes. What did you think of it?
Utterance 4:	I thought it was better than the others. I really liked it. What did you think of it?
Responses	
Reference	Yes, I liked it too. It was exciting but not over the top. Do you know what I mean?
HRED [8]	It was a good game.
WSeq [9]	I am sorry that you have been using it for two weeks without paying.
SD-HRED [10]	I was sitting at the door. I was on the opening at the TV station.
HRAN [11]	I am not sure I will be able to find a job.
ReCoSa [12]	I like it.
KS-CQ(ours)	I liked it too. I thought it was a good movie.

as a feedback to it. WSeq, a method proposed in [9], also considers to determine the importance of each context by its similarity to the query. However, WSeq is easily misled to choose an irrelevant context and generate an inappropriate response (see Table I), due to words that frequently appear in the context without carrying a specific meaning, e.g., "did," "you," and "it."

To address the above challenges, we propose a context modeling framework named KS-CQ for the multi-turn response generation task. It contains two pivotal components, namely, the Keep and the Select modules, to produce a neighbor-aware Context representation and a context-enriched Query representation. The Keep module focuses on explicit responds-to relationships and recodes each utterance in context by a transformer encoder structure to use semantic information from its neighbor utterances, i.e., prior and posterior adjacent utterances. Then, through a sequential across-turn integration, we obtain a neighbor-aware context representation that not only contains semantics of utterances but also captures the discourse coherence within context. The Select module first uses self-attention to capture relationships between words of a query and then adopts context attention to let the query select relevant context and thereby enrich itself. With the Keep and the Select modules, the proposed KS-CQ model can generate a contextually coherent response with presenting more informativeness, as shown in Table I.

To examine the effectiveness of our proposal and its pivotal components, we conduct extensive experiments on two benchmark datasets. Our experiments show that with the proposed Keep and the Select modules, our KS-CQ model outperforms the state-of-the-art baselines in terms of automatic and human evaluations, demonstrating its effectiveness on generating appropriate and informative responses. We also analyze the impact of context length and query length on response generation in the KS-CQ model, finding that it is robust to contexts and queries of various lengths, especially presents strong performance on cases with short queries.

Our main contributions can be summarized as follows.

1) We propose to represent conversational context by accounting for explicit responds-to relationships between adjacent utterances and addressing the dominant role of the query in response generation.
2) We propose a novel hierarchical context modeling framework named KS-CQ for multi-turn response generation task, which mainly consists of a Keep module to produce a neighbor-aware context representation and a Select module to generate context-enriched query representation.
3) We conduct extensive experiments on two benchmark conversational datasets to examine the effectiveness of our proposal and its pivotal components, finding it outperforms the state-of-the-art baselines in terms of both automatic and human evaluations.

Next, we review related work in Section II and then present the details of the KS-CQ model in Section III. In Section IV, we describe our experimental settings. Section V presents the experimental results and reflections on the outcomes. We formulate our conclusions in Section VI.

II. RELATED WORK

We mainly review two types of related work: open-domain conversational systems and multi-turn response generation.

A. Open-Domain Conversational Systems

The conversational systems can be classified into two types, i.e., task-oriented and non-task-oriented (or open-domain) [2], [3]. The task-oriented conversational systems [23] are designed to help users complete specific tasks, e.g., searching products and booking flights. The open-domain conversational systems [1] focus on realizing natural interactions with humans on open-domain topics. As pointed by Huang *et al.* [1], the goal of an open-domain system is to ensure long-term user engagement, which is difficult to optimize for and requires a comprehensive understanding of the conversational context to produce appropriate responses.

Three kinds of approach have been developed for open-domain conversational systems, including generation-based, retrieval-based, and hybrid. In the generation-based methods [4]–[6], [9]–[12], [16], [17], the responses are generated word by word in an auto-regressive manner. Inspired by statistical machine translation, the encoder-decoder framework [15] is the most popular choice for neural generative conversational models, where conversational context is first encoded into semantic vectors and the decoder takes the context representation as input to sample a word from a pre-defined vocabulary. Conditional variational autoencoders [19] and generative adversarial networks [24] are also used. With the success of transformer framework [25], pre-trained generative models on large-scale datasets, such as GPT-2 [26], GPT-3 [27] and DialoGPT [6], can produce responses closely emulating real-world text written by humans. As for the retrieval-based methods [28], [29], given a conversational context, they select a response from a pre-defined corpus of candidate responses. The key to response selection is the matching process between the response and the context.

In general, the generation-based methods tend to provide flexible but generic responses, while the retrieval-based methods can give informative but blunt responses [3]. A natural way for performance improvement is to assemble them in a unified framework, i.e., hybrid conversational models [30], [31], where retrieved candidate responses together with the conversation context are input to a neural response generator, and the final response is produced by a post-ranker.

Our work investigates the generation-based approach for the open-domain conversational system. To make an engaging conversation, we propose a novel context modeling method to perform a deep understanding of conversation content through capturing discourse coherence and ongoing topic.

B. Multi-Turn Response Generation

Response generation is at the heart of an open-domain conversational system [1]. In a multi-turn scenario, responses need to be coherent and consistent with respect to the conversation context. From the perspective of context modeling, the current multi-turn response generation methods can be grouped into two major types: non-hierarchical and hierarchical.

In non-hierarchical response generation, all historical utterances are processed as a whole and a single-turn conversational framework is adopted to produce a response. Early non-hierarchical models [4], [5], [32] produce a concatenation of the original contextual utterances or their corresponding encoded vector representations, and then use an RNN-based encoder-decoder network [15] for response generation. Transformers [25] present a more powerful architecture than RNNs for modeling long sequences and have been applied to multi-turn response generation in some studies, e.g., DialoGPT [6], T5 [33], and GPT-3 [27]. Besides the above concatenation strategy, some studies [7], [13] propose rewriting mechanisms to model contextual utterances, where a user's latest utterance will be rewritten into a new utterance by restoring omitted information and co-references. However, the non-hierarchical models ignore semantic relationships within contextual utterances, which actually provide rich information about the dynamic conversation flow across multi-turn interactions [3], [16].

For hierarchical response generation, Serban *et al.* [8] first proposed a hierarchical framework, HRED, with two-level recurrent encoders to generate the context representation by integrating individual utterance embeddings. It gives rise to new insights about modeling context at multiple semantic levels. Since then, Serban *et al.* [18] introduced a latent variable model into HRED to improve the response diversity. Chen *et al.* [17] used a memory network to enhance HRED in terms of modeling long-term dependencies. Considering that response is only related to a few previous contextual utterances, some researchers attempted to investigate the relevance between the context and the response. For instance, Tian *et al.* [9] proposed a model named WSeq that emphasizes the significance of the similarity between the context and the user's latest utterance. The SD-HRED model proposed by Zhang *et al.* [10] concentrates on weighting the importance of each contextual utterance with static and dynamic attentions. Xing *et al.* [11] argued that words and utterances in context have different degrees of importance for response generation and then presented the HRAN method to model the hierarchy and levels of importance. Zhang *et al.* [12] combined transformer [25] and HRED, proposing the ReCoSa method to leverage masked response representation to detect relevant contexts.

Our work follows the hierarchical manner of context modeling. Unlike previous hierarchical studies, which encode utterances individually and apply an integration strategy to fuse them afterward, our work considers the responds-to relationship between adjacent contextual utterances to optimize the representation learning of context. We also address the dominant role of the query, i.e., the user's latest utterance, in response generation, and design a context-enriched representing method to complement query with context.

III. Approach

We first formalize the multi-turn response generation task. Given M ($M \geq 2$) utterances of a conversation session $\{U_1, \ldots, U_M\}$, the last utterance U_M is denoted as query and $U_{<M} = \{U_1, \ldots, U_{M-1}\}$ is viewed as context. Thus, the purpose of the response generation task is to produce a response U_{M+1} given the context and the query by calculating a conditional probability $P(U_{M+1} \mid U_{<M}; U_M)$. Assuming that U_{M+1} is a sequence of N_{M+1} words, i.e., $(w_{1,M+1}, \ldots, w_{N_{M+1},M+1})$, the probability of generating a response can be decomposed as

$$P(U_{M+1} \mid U_{<M}; U_M) = \prod_{n=1}^{N_{M+1}} P\left(w_{n,M+1} \mid w_{<n,M+1}; U_{<M}; U_M\right) \quad (1)$$

where $w_{<n,M+1}$ denotes all previously generated words before nth step of U_{M+1}.

Our solution for the multi-turn response generation task is outlined in Fig. 1. The proposed KS-CQ model consists of four main components, i.e., the utterance encoder (see Section III-A), the Keep (see Section III-B) and the Select (see Section III-C) modules, and the response decoder (see Section III-D). For utterance encoding, we use a bidirectional recurrent structure to encode each utterance in context and query into a sequence of hidden vectors. Next, the Keep module recodes each utterance of context, i.e., $U_m \in U_{<M}$, by making it attentively keep semantics from its neighboring utterances, including the prior and the posterior adjacent utterances, i.e., U_{m-1} and U_{m+1}. Through an across-turn integration, the Keep module can produce a neighbor-aware context representation. After that, the Select module computes within-query relevance to make each word of U_M absorb information from other words, and then selectively introduces relevant semantics from context to obtain a context-enriched query representation. Finally, the response U_{M+1} is generated by a recurrent decoder with attentively taking the neighbor-aware context representation and context-enriched query representation as input.

A. Utterance Encoder

The utterance encoder aims to obtain the initial word-level embedding of each individual utterance in both context

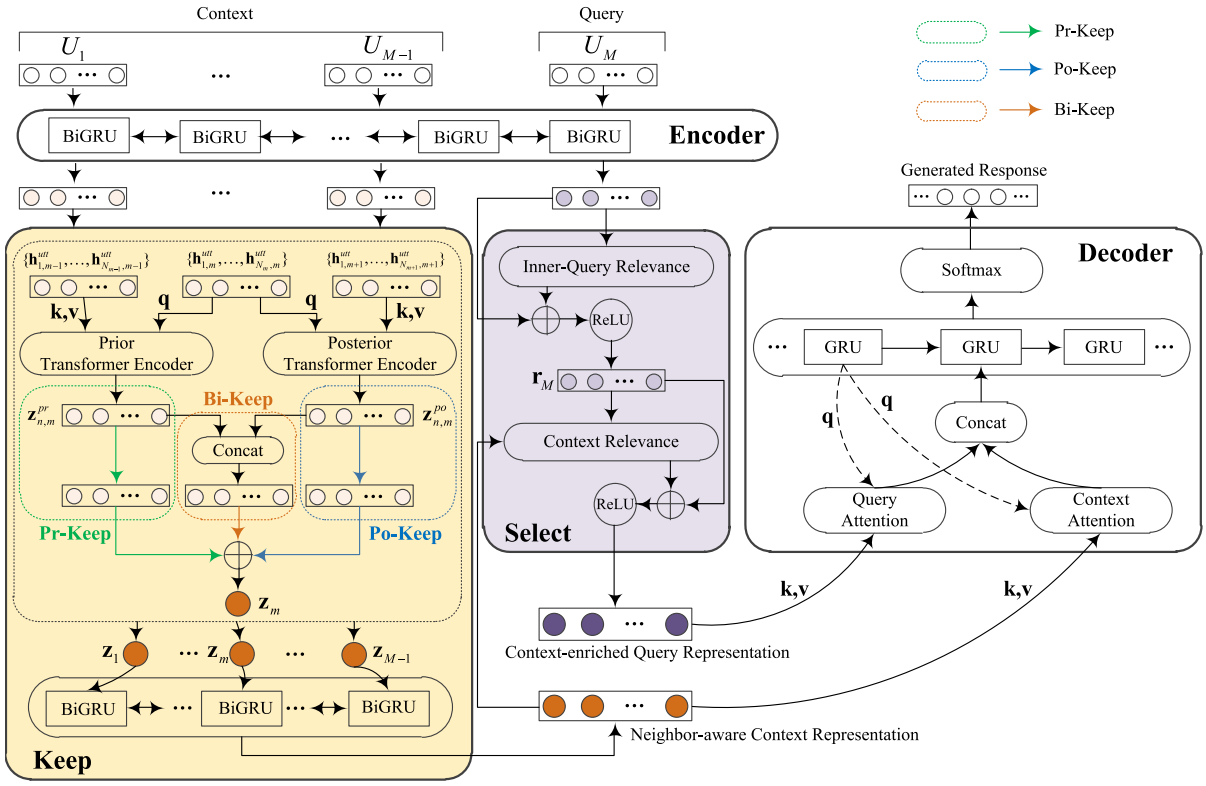


Fig. 1. Overview of the KS-CQ model.

and query. Given M utterances of a conversation session, i.e., $\{U_1, \ldots, U_M\}$, each utterance U_m ($m \in [1, M]$) contains a sequence of N_m words, denoted as $U_m = \{w_{1,m}, \ldots, w_{N_m,m}\}$. We apply a bidirectional gated recurrent unit (BiGRU) [34] to convert each word $w_{n,m}$ ($n \in [1, N_m]$) into a hidden vector as follows:

$$\overrightarrow{\mathbf{h}^{\text{utt}}_{n,m}} = \overrightarrow{\text{BiGRU}_u}\left(\overrightarrow{\mathbf{h}^{\text{utt}}_{n-1,m}}, \mathbf{e}_{w_{n,m}}\right) \tag{2}$$

$$\overleftarrow{\mathbf{h}^{\text{utt}}_{n,m}} = \overleftarrow{\text{BiGRU}_u}\left(\overleftarrow{\mathbf{h}^{\text{utt}}_{n-1,m}}, \mathbf{e}_{w_{n,m}}\right) \tag{3}$$

and then

$$\mathbf{h}^{\text{utt}}_{n,m} = \overrightarrow{\mathbf{h}^{\text{utt}}_{n,m}} + \overleftarrow{\mathbf{h}^{\text{utt}}_{n,m}} \tag{4}$$

where $\mathbf{e}_{w_{n,m}}$ is the randomly initialized word embedding of $w_{n,m}$; $\overrightarrow{\mathbf{h}^{\text{utt}}_{n,m}}$ and $\overleftarrow{\mathbf{h}^{\text{utt}}_{n,m}}$ are the respective hidden vectors of $w_{n,m}$ for the forward and backward passes. By the utterance encoder, U_m is represented as a sequence of hidden vectors, i.e., $\{\mathbf{h}^{\text{utt}}_{1,m}, \ldots, \mathbf{h}^{\text{utt}}_{N_m,m}\}$, which is denoted as the initial word-level embedding of utterance U_m.

B. Keep Module

In a multi-turn conversation, contextual utterances are interdependent as they follow local discourse coherence and thus may be linked by the responds-to relationships. Hence, we recode each utterance of context by introducing relevant semantics from its neighbors, including the prior and the posterior adjacent utterances. Given the initial word-level embedding of utterance U_m in context, i.e., $\{\mathbf{h}^{\text{utt}}_{1,m}, \ldots, \mathbf{h}^{\text{utt}}_{N_m,m}\}$ ($m \in [1, M]$), we use a transformer encoder [25] to inject the relevant information from its neighbor utterances, i.e., U_{m-1} and U_{m+1} into U_m. It is worthy noting that the neighbor-aware recoding procedure contains two transformer encoders from different directions, namely, a prior one and a posterior one.

Formally, for each word $w_{n,m}$ ($n \in [1, N_m]$) of U_m, the prior transformer encoder $\overleftarrow{\text{TF}}$ takes its hidden vector $\mathbf{h}^{\text{utt}}_{n,m}$ as the input query vector of attention, and the hidden vectors of U_{m-1} as key and value vectors as follows:

$$\begin{cases} \mathbf{z}^{\text{pr}}_{n,m} = \overleftarrow{\text{TF}}(\mathbf{q}, \mathbf{k}, \mathbf{v}), m \in (1, M) \\ \mathbf{q} = \mathbf{h}^{\text{utt}}_{n,m} \\ \mathbf{k}, \mathbf{v} \in \left\{\mathbf{h}^{\text{utt}}_{1,m-1}, \ldots, \mathbf{h}^{\text{utt}}_{N_{m-1},m-1}\right\} \end{cases} \tag{5}$$

and for the first utterance U_1 in context, we will get

$$\mathbf{q}, \mathbf{k}, \mathbf{v} \in \left\{\mathbf{h}^{\text{utt}}_{1,1}, \ldots, \mathbf{h}^{\text{utt}}_{N_1,1}\right\}, \quad m = 1 \tag{6}$$

which turns to be a self-attention. For the posterior transformer encoder $\overrightarrow{\text{TF}}$, we have

$$\begin{cases} \mathbf{z}^{\text{po}}_{n,m} = \overrightarrow{\text{TF}}(\mathbf{q}, \mathbf{k}, \mathbf{v}), \quad m \in [1, M) \\ \mathbf{q} = \mathbf{h}^{\text{utt}}_{n,m} \\ \mathbf{k}, \mathbf{v} \in \left\{\mathbf{h}^{\text{utt}}_{1,m+1}, \ldots, \mathbf{h}^{\text{utt}}_{N_{m+1},m+1}\right\}. \end{cases} \tag{7}$$

After obtaining $\mathbf{z}^{\text{pr}}_{n,m}$ and $\mathbf{z}^{\text{po}}_{n,m}$, which denote representations of $w_{n,m}$ enhanced by the prior and the posterior neighbor utterances, respectively, we concatenate them as

$$\mathbf{z}_{n,m} = \text{Concat}\left[\mathbf{z}^{\text{pr}}_{n,m}; \mathbf{z}^{\text{po}}_{n,m}\right] \tag{8}$$

where $\mathbf{z}_{n,m}$ is the final representation for word $w_{n,m}$ of U_m, which not only contains its own meaning but also carries relevant semantics from neighbor utterances.

We denote the above operation as Bi-Keep, where $\mathbf{z}_{n,m}$ is obtained by a concatenation of $\mathbf{z}^{\text{pr}}_{n,m}$ and $\mathbf{z}^{\text{po}}_{n,m}$. We also present

two variants, i.e., Pr-Keep and Po-Keep, to give insights into the impact of neighbor utterances that are from different directions. Specifically, the Pr-Keep module only introduces relevant semantics from its prior neighbor utterance, which means $\mathbf{z}_{n,m} = \mathbf{z}^{\text{pr}}_{n,m}$. While the Po-Keep focuses on its posterior neighbor utterance, where $\mathbf{z}_{n,m} = \mathbf{z}^{\text{po}}_{n,m}$.

In this manner, we obtain a new sequence of N_m hidden vectors for each contextual utterance U_m, denoted as $\{\mathbf{z}_{1,m}, \ldots, \mathbf{z}_{N_m,m}\}$. We sum the N_m hidden vectors as follows:

$$\mathbf{z}_m = \sum_{n=1}^{N_m} \mathbf{z}_{n,m} \tag{9}$$

where $\mathbf{z}_m$ denotes a unified utterance-level representation of U_m, which contains neighbor-aware semantics from all its words. Then, to make each contextual utterance aware of its position in context, we conduct a discourse-level integration through a BiGRU structure as

$$\begin{cases} \overleftarrow{\mathbf{z}^{\text{keep}}_m} = \overleftarrow{\text{BiGRU}_k}\left(\overleftarrow{\mathbf{z}^{\text{keep}}_{m-1}}, \mathbf{z}_m\right) \\ \overrightarrow{\mathbf{z}^{\text{keep}}_m} = \overrightarrow{\text{BiGRU}_k}\left(\overrightarrow{\mathbf{z}^{\text{keep}}_{m-1}}, \mathbf{z}_m\right) \\ \mathbf{z}^{\text{keep}}_m = \overleftarrow{\mathbf{z}^{\text{keep}}_m} + \overrightarrow{\mathbf{z}^{\text{keep}}_m} \end{cases} \tag{10}$$

where $\mathbf{z}^{\text{keep}}_m$ denotes the discourse-level representation of U_m ($m \in [1, M-1]$), and $\{\mathbf{z}^{\text{keep}}_1, \ldots, \mathbf{z}^{\text{keep}}_{M-1}\}$ can be regarded as the neighbor-aware context representation.

C. Select Module

For the task of response generation in multi-turn conversations, the query, i.e., user's latest utterance, plays a dominant role because the response is produced as a feedback to it. Thus, instead of directly incorporating all context for response generation, we propose to process context as background information of the conversation and use it to enrich the query representation.

The Select module aims to make the query selectively absorb relevant semantic information from the neighbor-aware context representation produced by the Keep module. Specifically, by the utterance encoder, the words in query U_M have been represented as $\{\mathbf{h}^{\text{utt}}_{1,M}, \ldots, \mathbf{h}^{\text{utt}}_{N_M,M}\}$. For each word $w_{i,M}$ in U_M, we first calculate the importance of other words in U_M relative to it as

$$\beta_{i,j} = \frac{\exp\left(f\left(g^{\beta,q}_{\text{relu}}\left(\mathbf{h}^{\text{utt}}_{i,M}\right), g^{\beta,k}_{\text{relu}}\left(\mathbf{h}^{\text{utt}}_{j,M}\right)\right)\right)}{\sum_{j'=1, j'\neq i}^{N_M} \exp\left(f\left(g^{\beta,q}_{\text{relu}}\left(\mathbf{h}^{\text{utt}}_{i,M}\right), g^{\beta,k}_{\text{relu}}\left(\mathbf{h}^{\text{utt}}_{j',M}\right)\right)\right)} \tag{11}$$

where $\beta_{i,j}$ is the importance of $w_{j,M}$ relative to $w_{i,M}$ and $j \neq i \in [1, N_M]$. $g^{\beta,q}_{\text{relu}}$ and $g^{\beta,k}_{\text{relu}}$ are full-connected networks with ReLU activation function; f is the attention function and is implemented by dot-product operation. Then we attentively accumulate information from other words to update the representation of $w_{i,M}$ as follows:

$$\mathbf{r}_{i,M} = g^{\text{norm}}_{\text{relu}}\left(\mathbf{h}^{\text{utt}}_{i,M} + \sum_{j=1, j\neq i}^{N_M} \beta_{i,j} * \mathbf{h}^{\text{utt}}_{j,M}\right) \tag{12}$$

where $g^{\text{norm}}_{\text{relu}}$ is a full-connected network with ReLU activation function to conduct normalization.

After introducing semantics from other words in the query, we continue to make $w_{i,M}$ selectively absorb information from relevant contexts. To be specific, we first calculate the relevance between $w_{i,M}$ and each contextual utterance U_m ($m \in [1, M-1]$) as

$$\varphi_{i,m} = \frac{\exp\left(f\left(g^{\varphi,q}_{\text{relu}}\left(\mathbf{z}^{\text{keep}}_m\right), g^{\varphi,k}_{\text{relu}}\left(\mathbf{r}_{i,M}\right)\right)\right)}{\sum_{m'=1}^{M-1} \exp\left(f\left(g^{\varphi,q}_{\text{relu}}\left(\mathbf{z}^{\text{keep}}_{m'}\right), g^{\varphi,k}_{\text{relu}}\left(\mathbf{r}_{i,M}\right)\right)\right)}. \tag{13}$$

Then we selectively introduce semantic information from contextual utterances according to their corresponding relevances to $w_{i,M}$ as follows:

$$\mathbf{r}^{\text{select}}_{i,M} = g^{\text{norm}}_{\text{relu}}\left(\mathbf{r}_{i,M} + \sum_{m=1}^{M-1} \varphi_{i,m} * \mathbf{z}^{\text{keep}}_m\right). \tag{14}$$

Here, we can represent the query U_M as $\{\mathbf{r}^{\text{select}}_{1,M}, \ldots, \mathbf{r}^{\text{select}}_{N_M,M}\}$, which is enriched by both the inner-query relationships and relevant context information. For simplicity, $\{\mathbf{r}^{\text{select}}_{1,M}, \ldots, \mathbf{r}^{\text{select}}_{N_M,M}\}$ is rewritten as $\{\mathbf{r}^{\text{select}}_1, \ldots, \mathbf{r}^{\text{select}}_{N_M}\}$.

D. Response Decoder

The response decoder aims to generate the response step-by-step in an auto-regressive way. At the tth step, the response decoder calculates

$$\mathbf{p}(\hat{w}_t) = \text{Softmax}\left(\mathbf{W}_{\text{dec}}\mathbf{h}^{\text{dec}}_t\right) \tag{15}$$

where $\mathbf{p}(\hat{w}_t)$ is the predicted vector of probabilities over all words in a pre-defined vocabulary V. $\mathbf{W}_{\text{dec}}$ is a projection matrix. $\mathbf{h}^{\text{dec}}_t$ is the hidden state of the tth step in the decoder, which is obtained by a GRU structure as

$$\begin{cases} \mathbf{h}^{\text{dec}}_0 = \mathbf{r}^{\text{select}}_{N_M} \\ \mathbf{h}^{\text{dec}}_t = \text{GRU}\left(\mathbf{h}^{\text{dec}}_{t-1}, \left[\mathbf{e}_{\hat{w}_{t-1}}, \mathbf{c}_t\right]\right) \end{cases} \tag{16}$$

where $\mathbf{e}_{\hat{w}_{t-1}}$ is the word embedding of $\hat{w}_{t-1}$. $\mathbf{c}_t$ can be regarded as a unified representation of conversation, which is obtained by joint attentions, namely, a query and a context attentions as follows:

$$\begin{cases} \mathbf{c}_t = \text{Concat}\left[\mathbf{c}^q_t; \mathbf{c}^c_t\right] \\ \mathbf{c}^q_t = \sum_{n=1}^{N_M} \rho^q_{t,n}\mathbf{r}^{\text{select}}_n \\ \mathbf{c}^c_t = \sum_{m=1}^{M-1} \rho^c_{t,m}\mathbf{z}^{\text{keep}}_m \\ \rho^q_{t,n} = \text{Softmax}\left(g^q_{\text{tanh}}\left(\mathbf{h}^{\text{dec}}_{t-1}, \mathbf{r}^{\text{select}}_n\right)\right) \\ \rho^c_{t,m} = \text{Softmax}\left(g^c_{\text{tanh}}\left(\mathbf{h}^{\text{dec}}_{t-1}, \mathbf{z}^{\text{keep}}_m\right)\right) \end{cases} \tag{17}$$

where $\rho^q_{t,n}$ and $\rho^c_{t,m}$ denote the importances that the word $w_{n,M}$ of query and the utterance U_m of context hold to the tth step of generated response, respectively. g^q_{tanh} and g^c_{tanh} are the full-connected networks with tanh activation function.

We write KS-CQ to denote the model with the above decoding strategy. To gain insight into the impact of context and query on response generation, we also consider another

strategy, that is, to obtain $\mathbf{c}_t$ only from the query representation. We write KS-Q to denote this variant, where $\mathbf{c}_t$ is obtained by an attention mechanism over the context-enriched query representation that is produced by the Select module, i.e., $\{\mathbf{r}_1^{\text{select}}, \ldots, \mathbf{r}_{N_M}^{\text{select}}\}$. The computation process is formalized as

$$\begin{cases} \mathbf{c}_t = \sum_{n=1}^{N_M} \rho_{t,n} \mathbf{r}_n^{\text{select}} \\ \rho_{t,n} = \text{Softmax}\big(g_{\tanh}^q\big(\mathbf{h}_{t-1}^{\text{dec}}, \mathbf{r}_n^{\text{select}}\big)\big). \end{cases} \tag{18}$$

It is worth noting that both KS-CQ and KS-Q adopt the Bi-Keep module.

E. Loss Function

Following previous response generation models [9]–[12], [16], [17], we use the cross-entropy loss function as

$$L_\Theta = -\frac{1}{N_{M+1}} \sum_{t=1}^{N_{M+1}} \mathbf{p}(w_t) \log \mathbf{p}(\hat{w}_t) \tag{19}$$

where $\mathbf{p}(w_t)$ is the one-hot vector over the vocabulary V, and $\mathbf{p}(\hat{w}_t)$ is the predicted vector of the word probability distribution. Θ denotes the trainable parameters of our model.

The training process of the KS-CQ model is outlined in Algorithm 1. We first randomly initialize the parameter set Θ, which mainly includes word embeddings, parameters of neural structures in utterance encoder (denoted as Enc), Keep module (denoted as Keep), Select module (denoted as Select), and response decoder (denoted as Dec). Then, given a conversation session $\{U_1, \ldots, U_M\} \in \mathcal{D}$ ($\mathcal{D}$ is the dataset), we use the utterance encoder and the Keep module to produce the neighbor-aware context representation $\{\mathbf{z}_1^{\text{keep}}, \ldots, \mathbf{z}_{M-1}^{\text{keep}}\}$ from Step 4 to Step 11. Based on this, the Select module further produces the context-enriched query representation $\{\mathbf{r}_1^{\text{select}}, \ldots, \mathbf{r}_{N_M}^{\text{select}}\}$ at Step 12. For the t-position of the response U_{M+1}, the decoder predicts the probability vector over all words in vocabulary at Step 15, and the loss is obtained by the cross-entropy loss function at Step 17. We average the loss across the length of response U_{M+1} at Step 19, and eventually use back propagation to update parameters in Θ.

Algorithm 1 Training Process of the KS-CQ Model

1: randomly initialize the parameters Θ.
2: **for** epoch in range(Epochs) **do**
3: **for** $\{U_1, \ldots, U_M\} \in \mathcal{D}$ **do**
4: **for** $U_m \in \{U_1, \ldots, U_{M-1}\}$ **do**
5: **if** $m = 1$ **then**
6: $\mathbf{z}_m^{keep} = \textbf{Keep}(\textbf{Enc}(m), \textbf{Enc}(U_{m+1}))$
7: **else**
8: $\mathbf{z}_m^{keep} = \textbf{Keep}(\textbf{Enc}(U_{m-1}), \textbf{Enc}(m), \textbf{Enc}(U_{m+1}))$
9: # detailed by Eq. 2-Eq. 10.
10: **end if**
11: **end for**
12: $\{\mathbf{r}_1^{select}, \ldots, \mathbf{r}_{N_M}^{select}\} = \textbf{Select}(\textbf{Enc}(U_M), \{\mathbf{z}_1^{keep}, \ldots, \mathbf{z}_{M-1}^{keep}\})$
13: # detailed by Eq. 11-Eq. 14.
14: **for** t in range(N_{M+1}) **do**
15: $\mathbf{p}(\hat{w}_t) = \textbf{Dec}(\{\mathbf{z}_1^{keep}, \ldots, \mathbf{z}_{M-1}^{keep}\}, \{\mathbf{r}_1^{select}, \ldots, \mathbf{r}_{N_M}^{select}\})$
16: # detailed by Eq. 15-Eq. 17.
17: $L(t) = -\mathbf{p}(w_t) \log \mathbf{p}(\hat{w}_t)$
18: **end for**
19: $L = \frac{1}{N_{M+1}} \sum_{t=1}^{N_{M+1}} L(t)$
20: use back propagation to update Θ.
21: **end for**
22: **end for**
23: **return** Θ.

IV. Experimental Setup

We list the following research questions to guide our experiments.

1) *RQ1:* Does KS-CQ outperform start-of-the-art baselines for response generation in terms of automatic evaluation?
2) *RQ2:* How do query and context affect the performance of response generation in KS-CQ?
3) *RQ3:* How do neighbor utterances (from different directions) affect the Keep module, respectively?
4) *RQ4:* How does KS-CQ perform in terms of human evaluation?
5) *RQ5:* What is the contribution of the proposed Keep and Select modules? Do they really help boost the performance of KS-CQ?
6) *RQ6:* What is impact of context length on the performance of KS-CQ?
7) *RQ7:* How does the length of the query utterance affect KS-CQ?

A. Datasets and Pre-Processing

We conduct experiments on two multi-turn conversation datasets.

1) *DailyDialog*[1] *[35]:* is collected from human-to-human talks in daily life, where utterances tend to be colloquial. It contains about 1.3k English conversation sessions covering various open-domain topics such as culture and education. We use the official training/validation/test splits, i.e., 11 118/1000/1000.
2) *KdConv*[2] *[36]:* is a Chinese conversation dataset that contains 4.5k conversation sessions from three domains, namely, film, music, and travel. Different from the DailyDialog dataset, each utterance in KdConv is related to certain knowledge triples. We randomly divide the dataset according to the ratio of 80%:10%:10% for training, validation, and test, respectively.[3]

To enrich the training samples, we pre-process the datasets as follows. The M-turn ($M \geq 2$) conversation involves M utterances, i.e., $\{U_1, \ldots, U_M\}$. At the mth ($2 \leq m < M$) turn, we denote U_m as the query, $U_{<m} = \{U_1, \ldots, U_{m-1}\}$ as context, and U_{m+1} as the ground-truth response. Similar to [8], [10], [12], we adopt truncations on samples, where the

[1]Available at http://yanran.li/dailydialog
[2]Available at https://github.com/thu-coai/KdConv
[3]We perform a random split since there is no official data split in the KdConv dataset.

TABLE II
DESCRIPTIVE STATISTICS OF THE PRE-PROCESSED DATASETS; CONTEXT LENGTH DENOTES THE NUMBER OF UTTERANCES CONTAINED IN THE CONTEXT, WHILE UTTERANCE LENGTH DENOTES THE NUMBER OF WORDS CONTAINED IN EACH UTTERANCE

Variable	DailyDialog		KdConv	
	Training	Test	Training	Test
#conversations	76,052	6,740	56,644	6,294
#vocabulary	18,018	6,193	26,002	20,220
avg. context length	5.01	4.92	8.84	8.88
avg. utterance length	12.56	12.62	11.85	11.88

maximum turn length of conversation is 15 and the maximum utterance length is 50. We obtain 76 052/6740 samples for training/testing in the DailyDialog dataset and 56 644/6294 in the KdConv dataset. Table II shows the major statistics about the pre-processed datasets.

B. Model Summary

Considering that our task is hierarchical context modeling for response generation, we compare the performance of KS-CQ against the following competitive baselines.

1) *HRED [8]:* The first hierarchical context modeling method for response generation, which uses an utterance-level RNN to encode utterances and a discourse-level RNN to sequentially integrate utterance embeddings into context representations.
2) *WSeq [9]:* An improved method based on HRED, which considers the similarity between context and query to selectively integrate utterance embeddings for response generation.
3) *SD-HRED [10]:* A hierarchical model that proposes static and dynamic attention mechanisms to measure context-to-context and context-to-response importance for response generation.
4) *HRAN [11]:* A method that uses both word-level and utterance-level attentions to produce context representation. It proposes that words in context may have different degrees of importance to response generation.
5) *ReCoSa [12]:* A hybrid model of transformer and HRED, where both the utterance-level and the discourse-level encoders are implemented by the self-attention mechanism. It also leverages masked response representation to detect relevant context and gains the state-of-the-art performance on the task of hierarchical context modeling.

All the baselines use the separate encoding way in the utterance embedding, which neglect the inner relationships within context. Moreover, except for WSeq, most baselines ignore the distinct role of the query utterance in response generation.

As for our models, besides KS-CQ, we also consider six variants, whose component details are provided in Table III.

1) KS-Q is used to investigate the impact of query on response generation. It only inputs the context-enriched query representation into the response decoder.
2) PrKS-CQ and PoKS-CQ are proposed to analyze the influences of prior and posterior neighbor utterances, which, respectively, apply Pr-Keep and Po-Keep operations in the Keep module.
3) SA-S-CQ and S-CQ are used to investigate the contribution of the Keep module. SA-S-CQ replaces the Keep module with a self-attention transformer [25], while S-CQ directly removes the Keep module and sums the initial word-level embedding of each utterance as its context representation.
4) K-CQ is used to examine the effectiveness of the Select module. It removes the Select module from KS-CQ, using the initial word-level embedding of query as query representation.

C. Implementation Details

For all models, the word embeddings are randomly initialized with a dimension of 512 and updated during training. The GRU and BiGRU units have a two-layer structure with 512 hidden cells. The number of heads in the transformer structure is 4. The parameters are optimized by the Adam Optimizer with a learning rate of 0.0001 and gradient clipping. We set the mini batch size as 64. We implement our models and baselines on the basis of code released by Lan *et al.* [37],[4] which uses the PyTorch framework and is trained on a workstation with a TITAN RTX GPU.[5]

D. Evaluation Methodology

Following prior work on response generation [8], [10], [12], we use both automatic and human evaluation metrics.

1) Automatic Evaluation: We adopt two types of standard metrics.

1) *Appropriateness-Based Metrics:* A common way to evaluate the appropriateness of a generated response is to compare it with the ground-truth response. BLEU [38] has been found to be inconsistent with human evaluation, hence we use embedding-based topic similarity metrics [10], [17], [39].
Average is computed as

$$\text{Average} = 1 - \cos(\mathbf{e}_r, \mathbf{e}_{\hat{r}}) \tag{20}$$

$$\mathbf{e}_r = \frac{\sum_{w \in r} \mathbf{e}_w}{|r|} \tag{21}$$

where $\mathbf{e}_r$ and $\mathbf{e}_{\hat{r}}$ denote the sentence-level embeddings of ground-truth response r and generated response $\hat{r}$, respectively. $\mathbf{e}_w$ is the embedding of word w in a response. It is noticeable that $\mathbf{e}_{\hat{r}}$ is computed the same as $\mathbf{e}_r$.
Extrema also uses the cosine distance like (20), while $\mathbf{e}_r$ is obtained by vector extrema. Here, at dth dimension of $\mathbf{e}_r$, its value e_{rd} is the most extreme value among all word vectors in the response r. The computation is formulated as

$$e_{\text{rd}} = \begin{cases} \max_{w \in r} e_{\text{wd}}, & \text{if } e_{\text{wd}} > |\min_{w' \in r} e_{w'd}| \\ \min_{w \in r} e_{\text{wd}}, & \text{otherwise} \end{cases} \tag{22}$$

where e_{wd} is the dth dimension of word embedding $\mathbf{e}_w$. Greedy does not calculate sentence-level embeddings. For

[4]The code is available at https://github.com/gmftbyGMFTBY/MultiTurnDialogZoo

[5]Our implementation is open-sourced at https://github.com/katherinelyx/KS-CQ

TABLE III

COMPONENT DETAILS OF KS-CQ AND ITS VARIANTS. THE ✓OF EACH COLUMN DENOTES THE CORRESPONDING COMPONENT IS EMPLOYED IN THE MODEL. "QUERY-ONLY" MEANS THE DECODER ONLY TAKES THE QUERY REPRESENTATION AS INPUT, WHILE "CONTEXT+QUERY" DENOTES THE INPUT OF THE DECODER IS THE CONCATENATION OF CONTEXT AND QUERY REPRESENTATIONS

Model	Utterance Encoder	Keep			Self-Attention Transformer	Select	Response Decoder	
		Pr-Keep	Po-Keep	Bi-Keep			Query-only	Context+Query
KS-CQ	✓			✓		✓		✓
KS-Q	✓			✓		✓	✓	
PrKS-CQ	✓	✓				✓		✓
PoKS-CQ	✓		✓			✓		✓
SA-S-CQ	✓				✓	✓		✓
S-CQ	✓					✓		✓
K-CQ	✓			✓				✓

each word $w \in r$, we greedily match it with a word $\hat{w} \in \hat{r}$ according to the cosine distance of their corresponding word embeddings, i.e., $\mathbf{e}_w$ and $\mathbf{e}_{\hat{w}}$. Then the greedy score is computed as

$$\text{Greedy} = \frac{\text{Greedy}(r, \hat{r}) + \text{Greedy}(\hat{r}, r)}{2} \tag{23}$$

$$\text{Greedy}(r, \hat{r}) = \frac{\sum_{w \in r} \max_{\hat{w} \in \hat{r}} (1 - \cos(\mathbf{e}_w, \mathbf{e}_{\hat{w}}))}{|r|} \tag{24}$$

$$\text{Greedy}(\hat{r}, r) = \frac{\sum_{\hat{w} \in \hat{r}} \max_{w \in r} (1 - \cos(\mathbf{e}_{\hat{w}}, \mathbf{e}_w))}{|\hat{r}|}. \tag{25}$$

These embedding-based metrics can measure the appropriateness from the perspective of semantic relevance, not just the word overlap.

In practice, for the DailyDialog dataset, we use the publicly available word vectors pre-trained on the Google News Corpus with the Word2Vec method. For the KdConv dataset, we use the word embeddings[6] released by Li *et al.* [40], which are pre-trained by Word2Vec with large-scale Sina Weibo Corpus. Words that are not included in the above corpora will be initialized with zero vectors.

2) *Informativeness-Based Metrics:* We use **H(w)**, i.e., the average trigram word entropy [17], [18], to measure the informativeness of the generated response. For the ith word $w_{i,j}$ in the jth generated response, $H(w_{i,j}) = -p(w_{i,j}|w_{i-2,j}; w_{i-1,j}) \log p(w_{i,j}|w_{i-2,j}; w_{i-1,j})$, where $p(w_{i,j}|w_{i-2,j}; w_{i-1,j})$ is approximated by the frequency of the trigram $\{w_{i-2,j}; w_{i-1,j}; w_{i,j}\}$ in the training corpus. Thus

$$\mathbf{H(w)} = \frac{1}{|\Omega|} \sum_{j=1}^{|\Omega|} \frac{1}{N_j} \sum_{i=1}^{N_j} H\left(w_{i,j}\right) \tag{26}$$

where $|\Omega|$ is the total number of generated responses, and N_j denotes the number of words contained in the jth generated response.

2) Human Evaluation: Following [22], [41], we randomly select 300 samples from the DailyDialog test set. We conduct human evaluation on the DailyDialog dataset, as it involves daily conversations that are easy to understand and do not require domain knowledge to make the judgment. For each test sample, we generate responses using our models and baselines based on the given conversational history (context and query). We invite four undergraduate students who are not involved with this work as human annotators.

Given these predicted responses and their corresponding conversational history, the annotators are asked to give a rating based on the following two criteria [42]: appropriateness measures how appropriate the generated response is for the given conversational history. It can be understood as semantically relevant and logically reasonable; informativeness measures how informative the generated response is. Generally, the more meaningful words contained in a response, the more informative it is. This metric can distinguish engaging responses from generic and dull ones, e.g., "Yeah," "I'm not sure."

For each metric, each annotator will give a score ranged from 1 to 5 based on how the generated response performs on it. Higher value denotes better performance. Then, for each model, we average the scores provided by on annotator and continue averaging across four annotators. The final averaged value is regarded as the model's evaluated result.

[6]https://github.com/Embedding/Chinese-Word-Vectors

V. RESULTS AND DISCUSSION

A. Performance on Automatic Evaluation

To answer RQ1 to RQ3, we examine the quality of the responses generated by our models and the baselines in terms of Average, Extrema, Greedy, and $H(w)$, respectively. We also conduct statistical significance tests on the pairwise differences of the best performer versus the best baseline. The results are presented in Table IV.

Let us first concentrate on RQ1. As shown in Table IV, KS-CQ outperforms the baselines in terms of all metrics on both the datasets; the improvement in terms of Average on the DailyDialog dataset is by a large margin. This confirms the significant improvement of our proposal for generating appropriate and informative responses for multi-turn conversations. Among the baselines, HRED shows superiority on appropriateness-based metrics in most cases and WSeq keeps consistently advantage on informativeness-based metric. Compared with our models, most baselines present inconsistent performance on different metrics. For instance, in the DailyDialog dataset, ReCoSa performs quite well on Greedy while gains the lowest value on $H(w)$. In the KdConv dataset, HRED is the best baseline in terms of Average, while its performance in terms of $H(w)$ is non-ideal. It indicates the difficulty of obtaining comprehensively good performance on response generation. In contrast, our KS-CQ model enable achieve balanced improvement on both response appropriateness and informativeness.

TABLE IV

MODEL PERFORMANCE ON AUTOMATIC EVALUATION. THE BEST PERFORMER AND THE BEST BASELINE OF EACH COLUMN ARE BOLDFACED AND UNDERLINED, RESPECTIVELY. STATISTICAL SIGNIFICANCE OF PAIRWISE DIFFERENCES OF THE BEST PERFORMER VERSUS THE BEST BASELINE IS DETERMINED BY A t-TEST ($^{\blacktriangle}$ FOR $\alpha = 0.05$)

Model	Average	Extrema	Greedy	H(w)
DailyDialog				
HRED	0.5846	0.7494	0.4769	12.8579
SD-HRED	0.5662	0.7436	0.4612	12.6604
WSeq	0.5766	0.7546	0.4393	14.1442
HRAN	0.5526	0.7469	0.4534	10.9604
ReCoSa	0.5152	0.6228	0.5239	8.3061
KS-CQ	**0.6441**$^{\blacktriangle}$	**0.7910**$^{\blacktriangle}$	**0.5277**	14.2509
KS-Q	0.6234	0.7729	0.5033	14.0409
PrKS-CQ	0.6313	0.7817	0.4955	**14.3298**$^{\blacktriangle}$
PoKS-CQ	0.6277	0.7763	0.4927	14.2724
KdConv				
HRED	0.7261	0.8495	0.5495	15.0060
SD-HRED	0.7125	0.8393	0.5511	15.0865
WSeq	0.7145	0.8418	0.5486	15.3035
HRAN	0.6401	0.8117	0.4592	14.5548
ReCoSa	0.7063	0.8382	0.5373	15.1624
KS-CQ	**0.7273**	**0.8512**	**0.5582**	**15.4352**$^{\blacktriangle}$
KS-Q	0.7215	0.8495	0.5339	15.3047
PrKS-CQ	0.7185	0.8486	0.5344	15.2958
PoKS-CQ	0.7130	0.8455	0.5267	15.2625

Besides, for all models, we see that the performance in terms of all metrics on the DailyDialog dataset is lower than that on the KdConv dataset. This may be attributed to the fact that the DailyDialog dataset is collected from human daily talks and contains many colloquial expressions. Most conversations in the KdConv dataset can be grounded to certain knowledge graphs, and thus utterances are usually more informative and more recognizable than those in the DailyDialog dataset. The larger improvements achieved by KS-CQ over the best baselines on the DailyDialog dataset demonstrate the ability of KS-CQ to model semantically sparse conversations. However, we can also note that the performance gains of our model on the KdConv dataset are mostly not significant. This indicates that our model still has improvement room on dealing with such knowledge-driven conversations.

Next, we move to RQ2. In Table IV, compared with KS-CQ, the variant model KS-Q that only uses the context-enriched query representation to generate a response displays performance drop in terms of all metrics on both the datasets. This demonstrates the importance of context to response generation in KS-CQ. Interestingly, even without inputting context to response decoder, KS-Q can beat some baselines on certain metrics, such as it outperforms all baselines in terms of Average and Extrema on the DailyDialog dataset. Moreover, among the baselines, we can see that HRED and WSeq, emphasizing the effect of the query on response generation, obtain relatively good performance. This confirms the importance of the query, and we conclude from the empirical test that: 1) context and query are both important to multi-turn response generation and 2) the query, i.e., the latest utterance, usually plays a dominant role in response generation and can help filter out noises in the context. The outstanding performance of KS-Q can be attributed to the Select module. It uses the context-enriched query representation to generate a response, which can absorb relevant contextual semantics even without directly taking the context as input.

TABLE V

MODEL PERFORMANCE IN TERMS OF HUMAN EVALUATION ON THE DAILYDIALOG DATASET. THE VALUE IN BRACKETS DENOTES THE STANDARD DEVIATION OF THE AVERAGED RESULTS FROM DIFFERENT ANNOTATORS. THE BEST PERFORMER AND THE BEST BASELINE OF EACH COLUMN ARE BOLDFACED AND UNDERLINED, RESPECTIVELY. STATISTICAL SIGNIFICANCE OF PAIRWISE DIFFERENCES OF KS-CQ VERSUS THE BEST BASELINE IS DETERMINED BY A t-TEST ($^{\blacktriangle}$ FOR $\alpha = 0.05$)

Model	Appropriateness	Informativeness
HRED	2.1537 ($\pm$0.1317)	2.5717 ($\pm$0.3617)
SD-HRED	2.7250 ($\pm$0.1717)	2.4917 ($\pm$0.3950)
WSeq	2.4883 ($\pm$0.2383)	2.4050 ($\pm$0.4950)
HRAN	1.7127 ($\pm$0.0960)	1.9283 ($\pm$0.0216)
ReCoSa	1.4440 ($\pm$0.0427)	1.2238 ($\pm$0.0505)
KS-CQ	**2.9383**$^{\blacktriangle}$ ($\pm$0.0403)	**2.8150**$^{\blacktriangle}$ ($\pm$0.2719)

Let us turn to RQ3. First, as shown in Table IV, in most cases KS-CQ achieves better performance than PrKS-CQ or PoKS-CQ, which only introduces prior or posterior neighbor utterances. Moreover, on the DailyDialog dataset, PrKS-CQ and PoKS-CQ beat the baselines on most metrics (except Greedy). This confirms the intuition underlying the Keep module that utterances in context are interdependent and their representations can be enhanced by neighboring utterances from both prior and posterior directions. Furthermore, PrKS-CQ consistently outperforms PoKS-CQ and even beats KS-CQ in terms of $H(w)$ on the DailyDialog dataset. We interpret this as saying that prior neighboring utterances may matter more than posterior ones, since the generation of a conversation is a sequential progression where prior utterances usually provide the background for posterior ones.

B. Performance on Human Evaluation

To answer RQ4, we conduct a human evaluation on the DailyDialog dataset in terms of appropriateness and informativeness. Statistical significance tests on the pairwise differences of the best performer versus the best baseline are presented. Besides, we also present the standard deviations of the averaged results from different annotators. The results of the human evaluation are listed in Table V.

As shown in Table V, KS-CQ achieves the best performance in terms of both appropriateness and informativeness, which confirms its effectiveness on response generation from a subjective view. Especially, compared with the best baselines, i.e., SD-HRED and HRED, KS-CQ gains lower standard deviations on both the metrics. This indicates that different annotators present relatively high consistency to the good performance of our model.

As to baselines, we can see that HRAN and ReCoSa perform non-ideally in human evaluation. A closer look at the samples generated by ReCoSa reveals that it fails to produce natural responses while it usually does provide one or several relevant keywords, which may explain its good performance on the (automatic) Greedy metric and its poor performance in human evaluation. Besides, on a scale of 5, no model achieves a high absolute score in human evaluation. It actually reflects the gap between current response generation models and what people expect.

TABLE VI

PERFORMANCE OF KS-CQ, S-CQ, SA-S-CQ, AND K-CQ ON AUTOMATIC EVALUATION. THE VALUE IN BRACKETS DENOTES THE PERFORMANCE DROP RATIO COMPARED WITH THE PROPOSED KS-CQ MODEL. "KEEP→SA" DENOTES REPLACING THE KEEP MODULE OF KS-CQ WITH A SELF-ATTENTION TRANSFORMER. STATISTICAL SIGNIFICANCE OF PAIRWISE DIFFERENCES OF THE BEST PERFORMER VERSUS THE SECOND BEST PERFORMER IS DETERMINED BY A t-TEST ($^\blacktriangle$ FOR $\alpha = 0.05$)

Model	Average	Extrema	Greedy	H(w)
DailyDialog				
KS-CQ	0.6441	0.7910	0.5277$^\blacktriangle$	14.2509
S-CQ (w/o Keep)	0.6391 (-0.78%)	0.7883 (-0.34%)	0.5008 (-5.10%)	14.4727$^\blacktriangle$ (+1.56%)
SA-S-CQ (Keep→SA)	0.6422 (-0.29%)	0.7830 (-1.01%)	0.5122 (-2.94%)	14.1823 (-0.48%)
K-CQ (w/o Select)	0.6348 (-1.44%)	0.7819 (-1.15%)	0.5021 (-4.85%)	14.1159 (-0.95%)
KdConv				
KS-CQ	0.7273	0.8512	0.5582$^\blacktriangle$	15.4352$^\blacktriangle$
S-CQ w/o Keep	0.7233 (-0.55%)	0.8502 (-0.12%)	0.5424 (-2.83%)	15.1448 (-1.88%)
SA-S-CQ (Keep→SA)	0.7172 (-1.39%)	0.8457 (-0.65%)	0.5404 (-3.19%)	14.5540 (-5.71%)
K-CQ w/o Select	0.7262 (-0.15%)	0.8518 (+0.07%)	0.5438 (-2.58%)	15.1482 (-1.86%)

C. Contributions of the Keep and Select Modules

To answer RQ5, we use three variant models, i.e., SA-S-CQ, S-CQ, and K-CQ, comparing them with the KS-CQ model in terms of automatic evaluation. The component details of variant models are shown in Table III. The experimental results are shown in Table VI.

As shown in Table VI, compared with KS-CQ, the performance of SA-S-CQ and S-CQ in terms of most metrics declines on both the datasets, validating the effectiveness of the Keep module. Then, we can also find performance drops of K-CQ. This indicates the contribution of the Select module. Furthermore, by comparing S-CQ and K-CQ on the DailyDialog dataset, we see that the performance drop of K-CQ is larger than that of S-CQ in terms of most metrics; it turns out to be opposite on the KdConv dataset. This may be attributed to the fact that the query utterances in the DailyDialog dataset are relatively colloquial and informal, due to frequent omission and co-references. Such condition emphasizes the function of the Select module, as it can selectively absorb relevant semantics from context to enrich itself. While in the KdConv dataset, although utterances usually contain informative entities, the context of conversations tends to be long, which highlights the importance of the Keep module as it provides a memorization ability to capture long-term dependencies.

When we combine the results of Tables VI and IV, we see that S-CQ and K-CQ beat the best baselines on several metrics; for example, both S-CQ and K-CQ achieve better performance in terms of Average and Extrema on DailyDialog compared with the best baselines, i.e., HRED and WSeq, respectively. In summary, both the Keep and the Select modules play key roles in KS-CQ, while they can also provide strong performance on response generation solely by themselves. A possible direction is to incorporate these two modules solely or together into other context modeling frameworks, which may bring more improvements.

D. Analysis of the Impact of Context Length

To answer RQ6, we analyze the performance of KS-CQ and baselines on the test samples with varying context lengths, i.e., the number of utterances contained in context. Due to space limitations, we only present our results on the DailyDialog dataset. We split these 6740 test samples into three groups according to their corresponding context length, and finally get 63.32% featuring the length ranged in [1,5], 28.95% in (5,10], and 7.73% larger than 10 (denoted as >10). Then we evaluate the model performance in terms of various metrics. The results are plotted in Fig. 2.

As shown in Fig. 2, KS-CQ consistently obtains the best performance in terms of Average and Extrema with varying context length, which demonstrates that it can generate appropriate responses for both short and long conversations. Although we find that for cases with context lengths larger than 5, KS-CQ loses to ReCoSa in terms of Greedy and to WSeq in terms of $H(w)$, these two baselines gain polarized performance. For example, ReCoSa performs well on Greedy while bad on $H(w)$, and WSeq is just the opposite. It shows that KS-CQ can keep a balanced performance on metrics of different perspectives, and such ability is also robust to the variations in context length.

E. Analysis of the Impact on Query Length

To answer RQ7, we conduct an analysis of the performance of KS-CQ and baselines on the test samples with varying query lengths, i.e., the number of words contained in the query utterance. Due to space limitations, we only present our results on the DailyDialog dataset. We split these 6740 test samples into three groups based on their query length, where 13.71% of the samples have a query length in [1,5], 60.28% in (5,15], and 26.01% bigger than 15 (denoted as >15). Generally, a bigger length indicates more information in the query. We evaluate the model performance in terms of various metrics. The results are plotted in Fig. 3.

As shown in Fig. 3, compared with the baselines, KS-CQ achieves consistently good performance in terms of all metrics with varying query lengths. With short queries that contain no more than five words (i.e., the [1,5] group), the performance gap between KS-CQ and the baselines is larger than that with long queries. This demonstrates that KS-CQ can make full use of the limited semantics carried by short queries so as to generate appropriate and informative responses. This may be attributed to the Select module, which helps enrich the query representation by selectively absorbing information from relevant contexts. However, as to baselines, they either neglect the distinct role of query or fail to deal with short query.

As the query length decreases, the performance of all models on all metrics presents a down-going trend. Thus, we hypothesize that short queries are usually more difficult for the response generation task, since in this condition only limited information is available to track the ongoing conversation focus. Moreover, short queries are often generic utterances, e.g., Yeah, OK, and modal particles like Umh.

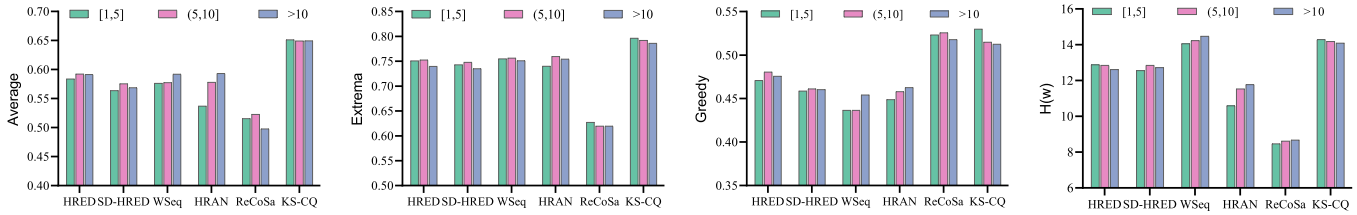


Fig. 2. Model performance under varying context lengths (the number of utterances in the context), where ■, ■ and ■ denote results with the context length in [1,5], (5,10], and >10, respectively.

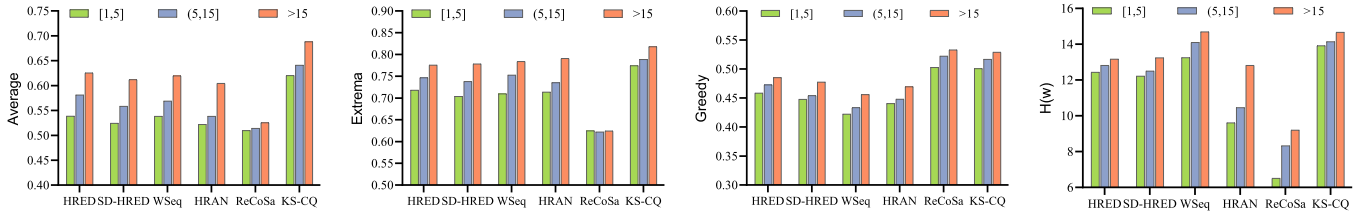


Fig. 3. Model performance under varying query lengths (the number of words in the query utterance), where ■, ■ and ■ denote results with the query length in [1,5], (5,15], and >15, respectively.

To gain insights into the model performance on such cases, we select samples that feature short queries with no more than five words from the DailyDialog test set. Then we conduct an analysis of the performance on these samples with varying context lengths, similar to Section V-D. We sample 942 such cases, where 66.23% features a context length in [1,5], 25.76% in (5,10], and 8.01% in >10. The results of the response generation task on this sample of short queries are presented in Fig. 4.

First, by comparing the results of Fig. 4 with those of Fig. 2, we see that all models display a performance drop for most context lengths in terms of all metrics. This indicates that it is growing harder for context understanding and response generation, when the query is short or lack efficient information. Furthermore, as the context length increases, KS-CQ shows an increase for the informativeness-based metric, i.e., $H(w)$, and a decrease on the majority appropriateness-based metrics, i.e., Average and Greedy. This may be attributed to the fact that, on one hand, more context can provide more semantic information to characterize the conversation, which, in turn, can help produce specific and diverse words in responses. On the other hand, a bigger context length indicates more frequent topic transitions in the conversation, which makes it harder to predict the topic currently being discussed for response generation. Short queries are usually too uninformative to help detect relevant context and filter out noise. KS-CQ consistently outperforms the baselines on the selected samples with short queries. This indicates that given a sufficiently long context, KS-CQ still can efficiently extract useful context and track the ongoing conversation focus from samples with a short query so as to generate appropriate and informative responses.

F. Case Study

To ground our understanding of the models discussed, we perform a case study on both the test sets. Table VII presents several examples of generated responses.

In Example 1, the context is short, and the query is ambiguous as a typical case of one-to-many, which means there may be multiple proper responses. Then, HRED, SD-HRED, and most of our models produce appropriate responses, showing better comprehension of the conversation than other models. Compared with HRED and SD-HRED, KS-CQ, S-CQ, SA-S-CQ, and KS-Q, which contain the keyword "hamburger" or "coffee," seem to be more logically reasonable. However, compared with KS-CQ and SA-S-CQ, KS-Q and S-CQ concurrently produce improper words like "champagne." This indicates that on one hand it will hurt the coherence of a generated response without taking its context into consideration, as context provides the conversation background; on the other hand, no matter the Keep module or a self-attention transformer can boost the response consistence, they can capture the conversation background through modeling the semantical relationships within context.

In Example 2, the context is long and the query involves a question. Of the baselines, HRED, WSeq, and HRAN provide obviously irrelevant responses, indicating a failure to understand the context and capture the topic being discussed. SD-HRED and ReCoSa get the point of "music," while they are weak on producing informative responses. Compared with the baselines, our models except for S-CQ show better coherence to the reference response with containing the key phrase "classical music." It may be attributed to that the Keep module performs a memorization on context through the neighbor-aware recoding process, which can alleviate the long-term dependency issue existing in long context.

In Example 3, based on the context we can see that the conversation involves "ordering a dessert," and the query focuses on "some pie." In such case, WSeq and HRAN give irrelevant responses, and HRED and SD-HRED are not consistent with context or the reference response. ReCoSa produces a relatively appropriate but ambiguous response. It may be attributed to that the baselines fail to efficiently use the information contained in query. However, most of our

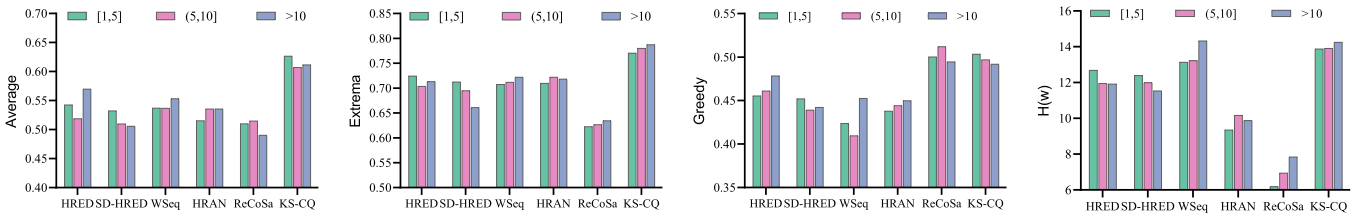


Fig. 4. Model performance on difficult samples with short queries ([1,5], i.e., containing no more than five words) under various context lengths, where ■, ■ and ■ denote results with context length ranged in [1,5], (5,10], and >10, respectively.

TABLE VII

EXAMPLES OF CONTEXT, QUERY, AND THE CORRESPONDING RESPONSES GENERATED BY VARIOUS MODELS. REFERENCE DENOTES THE GROUND-TRUTH RESPONSE IN THE DATASETS. "→" SEPARATES DIFFERENT CONVERSATION TURNS

	Context & Query	**Response**
1	**Context:** I would like to order breakfast please. → **Query:** What would you like madam?	**Reference:** Two muffins and a cup of coffee. Will it take long?
		HRED: I would like a bottle of champagne, lobster tail and filet mignon. **SD-HRED:** I would like to have a glass of tomato juice please. **WSeq:** Thanks. **HRAN:** Thank you. **ReCoSa:** I would like.
		KS-CQ: I would like a hamburger please. **S-CQ:** I would like a hamburger and a bottle of champagne, please. **SA-S-CQ:** I would like to have a coffee. **K-CQ:** I would like to have a bottle of wine and salad, please. **KS-Q:** I would like a hamburger and a bottle of champagne lobster tail and orange juice and chicken soup. **PrKS-CQ:** I would like to buy a double room for me please. **PoKS-CQ:** I would like to have a glass of champagne.
2	**Context:** What are you listening to? Is that Beethoven or Mozart? → It is Beethoven. Do you like it? → I think Beethoven's music is incredible. I have heard that listening to it can make you more intelligent too. Do you believe that? → I do not know about that but I do think that it helps people relax. → **Query:** What other kind of music do you listen to?	**Reference:** Actually I mostly just listen classical music. What about you?
		HRED: I do not know. I am a little bit of a. **SD-HRED:** I like the music too. **WSeq:** It is a pleasure. I have to go out of it. **HRAN:** I think I will have to discuss the possibility. **ReCoSa:**I like music.
		KS-CQ: I like classical music better. What do you think? **S-CQ:** Well, I like my Chinese hobbies, sometimes do it. **SA-S-CQ:** I enjoy listening to classical music and reading. **K-CQ:** I like a Chinese diet of classical music, but I do not eat spicy food. What are you going to do? **KS-Q:** I like classical music. **PrKS-CQ:** I heard that many people who play badminton too, but sometimes I play classical music too. **PoKS-CQ:** Not really, I like classical music.
3	**Context:** I want something sweet after dinner → What do you have in mind? → A dessert sounds nice. → What kind are you thinking of getting? → **Query:** I want to get some pie.	**Reference:** What kind of pie do you want?
		HRED: Do you want to get a burger? **SD-HRED:** I am sorry. I can not do that. **WSeq:** What do you want to go? **HRAN:** Thank you. **ReCoSa:** What do you want?
		KS-CQ: What kind of pie? Do you like to eat chocolate? **S-CQ:** What do you want to eat? **SA-S-CQ:** Are you kidding? **K-CQ:**What kind do you want to spend? **KS-Q:** What kind of food do you want? **PrKS-CQ:** What kind of drink? **PoKS-CQ:** What do you want?
4	**Context:** 连姆·尼森这个人你听说过吗? (Have you heard of Liam Neeson?) → 听说过啊, 他是英国的影视演员. (Yes, he is a British actor.) → 他的外文名叫什么呢? (What is his English name?) → 他的外文名字叫 Liam Neeson. (His English name is Liam Neeson.) → 他的星座是白羊座的吗? (Is he Aries?) → 不是的, 他是双子座的. 你知道他的妻子是谁吗? (No, his constellation is Gemini. Do you know his wife ?) → **Query:** 他的妻子是娜塔莎·理查德森. (His wife is Natasha Richardson.)	**Reference:** 他的个头是不是很高呀? (Is he tall?)
		HRED: 他是射手座的吗? (Is he a Sagittarius?) **SD-HRED:** 他都有哪些职业呢? (What is his occupation?) **WSeq:** 你知道他是哪个学校毕业的吗? (Do you know that what school he graduated from?) **HRAN:** 他是美国人. (He is an American.) **ReCoSa:** 哦, 那他的职业是什么? (Oh, what is his occupation?)
		KS-CQ: 哦, 他是个很优秀的演员. 你知道他是哪个学校毕业的吗? (Oh, he is an excellent actor. Do you know that which school he graduated from?) **S-CQ:** 你知道他的外文名字吗? (Do you know his English name?) **SA-S-CQ:** 好的, 谢谢你。(OK, thank you.) **K-CQ:** 他是哪个学校毕业的呢? (Do you know that which school he graduated from?) **KS-Q:** 他有外文名字吗? (Does he have an English name?) **PrKS-CQ:** 他的外文名字叫什么? (What is his English name?) **PoKS-CQ:** 嗯, 她的星座是射手座吗? (Em, is she a Sagittarius?)

models can provide relevant responses, especially the KS-CQ model accurately hits the keyword "pie." We attribute such performance to the Select module of our model which helps track the ongoing topics by the query utterance.

Example 4 is a representative case for the KdConv dataset, where we find that the conversation is driven by domain knowledge and its local topics transit across different turns, especially the reference response presents a sharp transition from the query. The responses generated by HRED, SD-HRED, HRAN, and ReCoSa seem reasonable and natural to the query, but they show repetition or some degree of conflict to the context. WSeq produces a relatively appropriate response. Of our models, KS-CQ, K-CQ, and PoKS-CQ also provide relevant responses, while KS-CQ also gives a good echo to the context "Yes, he is a British actor" by the expression "he is an excellent actor." Such phenomena may be ascribed to the Keep module that helps memorize the long context. However, we can note that responses produced by all models fail to perform consistence with the reference response. It may be attributed to that there is a sharp topic transition from the query to the reference response, which increases the difficulty of capturing the ongoing topic.

VI. Conclusion

In this article, we have improved the multi-turn context modeling in response generation by addressing two crucial factors, namely, the explicit responds-to relationship within context and the distinctive importance of the query utterance. We have proposed a neural response generation model, KS-CQ, which consists of two pivotal modules, i.e., the Keep and the Select modules. The Keep module recodes each utterance in the context by incorporating relevant semantics from its prior and posterior neighbor utterances, leading to a neighbor-aware context representation. The Select module focuses on making the query selectively absorb information from its context, leading to a context-enriched query representation. Extensive experiments conducted on two benchmark datasets confirm the effectiveness of the proposed KS-CQ model; it consistently outperforms competitive baselines in terms of automatic and human evaluations.

The proposed Keep and Select modules are possible to be incorporated solely or together into other multi-turn conversation models, which can provide strong ability of context modeling and further boost the response quality. However, our model also has limitation on modeling extreme long conversations, especially those featuring frequent topic transitions. As to future work, we are interested in introducing topic maintenance in the context modeling procedure, which aims to capture the topic flows underlying multi-turn interactions and implement reasonable topic transition in response generation. Our work can also be extended to multi-party conversations by introducing an additional detector so as to identify multi-threaded responds-to relationships.

References

[1] M. Huang, X. Zhu, and J. Gao, "Challenges in building intelligent open-domain dialog systems," *ACM Trans. Inf. Syst.*, vol. 38, no. 3, pp. 1–32, Jun. 2020.

[2] J. Gao, M. Galley, and L. Li, "Neural approaches to conversational AI," in *Proc. 41st Int. ACM Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2018, pp. 1371–1374.

[3] H. Chen, X. Liu, D. Yin, and J. Tang, "A survey on dialogue systems: Recent advances and new frontiers," *ACM SIGKDD Explor. Newslett.*, vol. 19, no. 2, pp. 25–35, Dec. 2017.

[4] A. Sordoni *et al.*, "A neural network approach to context-sensitive generation of conversational responses," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2015, pp. 196–205.

[5] L. Shang, Z. Lu, and H. Li, "Neural responding machine for short-text conversation," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics, 7th Int. Joint Conf. Natural Lang. Process.*, 2015, pp. 1577–1586.

[6] Y. Zhang *et al.*, "DIALOGPT: Large-scale generative pre-training for conversational response generation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics, Syst. Demonstrations*, 2020, pp. 270–278.

[7] H. Su *et al.*, "Improving multi-turn dialogue modelling with utterance ReWriter," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 22–31.

[8] I. V. Serban, A. Sordoni, Y. Bengio, A. C. Courville, and J. Pineau, "Building end-to-end dialogue systems using generative hierarchical neural network models," in *Proc. 13th Conf. Artif. Intell. (AAAI)*, 2016, pp. 3776–3784.

[9] Z. Tian, R. Yan, L. Mou, Y. Song, Y. Feng, and D. Zhao, "How to make context more useful? An empirical study on context-aware neural conversational models," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 231–236.

[10] W. Zhang *et al.*, "Context-sensitive generation of open-domain conversational responses," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 2437–2447.

[11] C. Xing, Y. Wu, W. Wu, Y. Huang, and M. Zhou, "Hierarchical recurrent attention network for response generation," in *Proc. 32nd Conf. Artif. Intell. (AAAI)*, 2018, pp. 5610–5617.

[12] H. Zhang, Y. Lan, L. Pang, J. Guo, and X. Cheng, "ReCoSa: Detecting the relevant contexts with self-attention for multi-turn dialogue generation," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 3721–3730.

[13] K. Zhou, K. Zhang, Y. Wu, S. Liu, and J. Yu, "Unsupervised context rewriting for open domain conversation," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1834–1844.

[14] Z. Pan, K. Bai, Y. Wang, L. Zhou, and X. Liu, "Improving open-domain dialogue systems via multi-turn incomplete utterance restoration," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1824–1833.

[15] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[16] C. Sankar, S. Subramanian, C. Pal, S. Chandar, and Y. Bengio, "Do neural dialog systems use the conversation history effectively? An empirical study," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 32–37.

[17] H. Chen, Z. Ren, J. Tang, Y. E. Zhao, and D. Yin, "Hierarchical variational memory network for dialogue generation," in *Proc. World Wide Web Conf. World Wide Web (WWW)*, 2018, pp. 1653–1662.

[18] I. V. Serban *et al.*, "A hierarchical latent variable encoder-decoder model for generating dialogues," in *Proc. 31st Conf. Artif. Intell. (AAAI)*. Palo Alto, CA, USA: AAAI Press, 2017, pp. 3295–3301.

[19] X. Shen *et al.*, "A conditional variational framework for dialog generation," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 504–509.

[20] T. Zhao, R. Zhao, and M. Eskénazi, "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 654–664.

[21] S. Vakulenko, M. de Rijke, M. Cochez, V. Savenkov, and A. Polleres, "Measuring semantic coherence of a conversation," in *Proc. 17th Int. Semantic Web Conf. (ISWC)*, 2018, pp. 634–651.

[22] W. Wang, M. Huang, X.-S. Xu, F. Shen, and L. Nie, "Chat more: Deepening and widening the chatting topic via a deep model," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jun. 2018, pp. 255–264.

[23] M. Henderson *et al.*, "Training neural response selection for task-oriented dialogue systems," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics*, 2019, pp. 5392–5404.

[24] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao, "Deep reinforcement learning for dialogue generation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1192–1202.

[25] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[26] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, p. 9, 2019.

[27] T. B. Brown *et al.*, "Language models are few-shot learners," *CoRR*, vol. abs/2005.14165, pp. 1–75, May 2020.

[28] J.-C. Gu, Z.-H. Ling, and Q. Liu, "Interactive matching network for multi-turn response selection in retrieval-based chatbots," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 2321–2324.

[29] J.-C. Gu, Z.-H. Ling, X. Zhu, and Q. Liu, "Dually interactive matching network for personalized response selection in retrieval-based chatbots," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1845–1854.

[30] D. Cai, Y. Wang, W. Bi, Z. Tu, X. Liu, and S. Shi, "Retrieval-guided dialogue response generation via a matching-to-generation framework," in *Proc. Conf. Empirical Methods Natural Lang. Process., 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 1866–1875.

[31] L. Yang *et al.*, "A hybrid retrieval-generation neural conversation model," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, Nov. 2019, pp. 1341–1350.

[32] O. Dušek and F. Jurčíček, "A context-aware natural language generator for dialogue systems," in *Proc. 17th Annu. Meeting Special Interest Group Discourse Dialogue*, 2016, pp. 185–190.

[33] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *CoRR*, vol. abs/1910.10683, pp. 1–67, Oct. 2019.

[34] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–Decoder approaches," in *Proc. 8th Workshop Syntax, Semantics Struct. Stat. Transl. (SSST)*, 2014, pp. 103–111.

[35] Y. Li, H. Su, X. Shen, W. Li, Z. Cao, and S. Niu, "DailyDialog: A manually labelled multi-turn dialogue dataset," in *Proc. 8th Int. Joint Conf. Natural Lang. Process.*, 2017, pp. 986–995.

[36] H. Zhou, C. Zheng, K. Huang, M. Huang, and X. Zhu, "KdConv: A Chinese multi-domain dialogue dataset towards multi-turn knowledge-driven conversation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 7098–7108.

[37] T. Lan, X. Mao, W. Wei, X. Gao, and H. Huang, "PONE: A novel automatic evaluation metric for open-domain generative dialogue systems," *CoRR*, vol. abs/2004.02399, pp. 1–37, Nov. 2020.

[38] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics (ACL)*, 2001, pp. 311–318.

[39] C.-W. Liu, R. Lowe, I. Serban, M. Noseworthy, L. Charlin, and J. Pineau, "How NOT to evaluate your dialogue system: An empirical study of unsupervised evaluation metrics for dialogue response generation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2122–2132.

[40] S. Li, Z. Zhao, R. Hu, W. Li, T. Liu, and X. Du, "Analogical reasoning on Chinese morphological and semantic relations," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 138–143.

[41] C. Xing *et al.*, "Topic aware neural response generation," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3351–3357.

[42] H. Song, Y. Wang, W.-N. Zhang, X. Liu, and T. Liu, "Generate, delete and rewrite: A three-stage framework for improving persona consistency of dialogue generation," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 5821–5831.

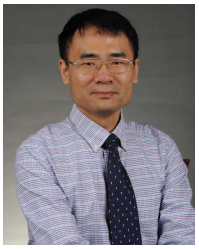
Yanxiang Ling received the M.S. degree in information system engineering from the National University of Defense Technology, Changsha, China, in 2013, where she is currently pursuing the Ph.D. degree.

She has published several papers at The Web Conference (WWW) and ACM International Special Interest Group on Information Retrieval Conference (SIGIR). Her research interests include neural response generation, question generation, and information retrieval.

Fei Cai received the Ph.D. degree in computer science from the University of Amsterdam, Amsterdam, The Netherlands, in 2015, under the supervision of Prof. Maarten de Rijke.

He is currently an Associate Professor with the National University of Defense Technology, Changsha, China. He has published several papers at ACM International Special Interest Group on Information Retrieval Conference (SIGIR), Conference on Information and Knowledge Management (CIKM), Foundations and Trends in Information Retrieval (FnTIR), *ACM Transactions on Information Systems* (TOIS), and IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE). His research interests include information retrieval and query formulation.

Dr. Cai serves as a PC Member for CIKM, SIGIR, ACM International Web-inspired research involving Search and Data Mining Conference (WSDM), and The Web Conference (WWW), as well as a Reviewer for SIGIR, WWW, WSDM, CIKM, IEEE TKDE, the *Journal of Information Processing and Management* (IPM), and the *Journal of the Association for Information Science and Technology* (JASIST).



Jun Liu (Senior Member, IEEE) received the B.S. degree in computer science and technology and the Ph.D. degree in systems engineering from Xi'an Jiaotong University, Xi'an, China, in 1995 and 2004, respectively.

He is currently a Professor with the Department of Computer Science, Xi'an Jiaotong University. He has authored more than 90 research papers in various journals and conference proceedings. His research interests include natural language processing (NLP), computer vision (CV), and e-learning.

Dr. Liu has won the Best Paper Awards in IEEE International Symposium on Software Reliability Engineering (ISSRE) 2016 and IEEE International Conference on Big Knowledge (ICBK) 2016. He also acted as the conference/workshop/track chair at numerous conferences. He has served as a Guest Editor for many technical journals, such as *Information Fusion* and IEEE SYSTEMS JOURNAL. He has been serving as an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (TNNLS) since 2020.

Honghui Chen received the Ph.D. degree in operational research from the National University of Defense Technology, Changsha, Hunan, China, in 2007.

He is currently a Professor with the National University of Defense Technology. He has published several papers at ACM International Special Interest Group on Information Retrieval Conference (SIGIR), the *Journal of Information Processing and Management* (IPM), and other top journals. His research interests include information systems and information retrieval.

Maarten de Rijke is currently a Distinguished University Professor of artificial intelligence and information retrieval with the University of Amsterdam, Amsterdam, The Netherlands, and the Scientific Director of the National Innovation Center for AI. With his team, he works on intelligent search, recommendation, and conversational assistance.

Prof. de Rijke is the former Editor-in-Chief of *ACM Transactions on Information Systems* and *Foundations and Trends in Information Retrieval*. He is the current Editor-in-Chief of the *Information Retrieval* book series.