Table-to-Dialog: Building Dialog Assistants to Chat With People on Behalf of You

HAIHONG E, ZECHENG ZHAN, AND MEINA SONG, (Member, IEEE)
School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

Corresponding author: Haihong E (ehaihong@bupt.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61902034.

ABSTRACT Artificial Intelligence (AI) personal assistant has attracted much attention from both academia and industry. Almost all existing AI personal assistants serve as service terminals to chat with human users for certain tasks. We are instead interested in building AI personal assistants for a different yet important dialog scenario, where they chat with people to fulfill specific tasks on behalf of their human users. As the personal assistants are playing a requester role, instead of a service terminal role, the conversation goal becomes delivering or requesting information according to specific user requests precisely and efficiently. The challenge for the conversation policy is that all user requests must be delivered precisely, while the challenge for the response generation is that it's generally expected for machine generated responses to cover multiple information slots, either requesting or delivering, to make the conversation efficient. In this paper, we present Table-to-Dialogue, a novel approach to address the above challenges when building a requester role AI personal assistant. We employ an encoder-decoder network to learn explicit conversation policy, which generates the corresponding information slots based on the conversation context and the user request table. We further integrate a novel Multi-Slot Constrained Bi-directional Decoder (MS-CBD) into the above encoder-decoder network, to generate machine response according to the multiple slot values and their intermediate representations from the policy decoder. Different from the existing single direction text decoder approaches, MS-CBD leverage the bi-directional context of the response when generating it to enhance the semantic coherence. The experiments shows that our approach significantly outperform the state-of-the-art conversation approaches on automatic and human evaluation metrics.

INDEX TERMS Dialogue system, sequence generation, deep learning, constrained decoding, table-to-text, natural language process.

I. INTRODUCTION

Building AI personal assistants is always a fascinating research topic ever since the middle of last century [1]–[3], [11]. Many successful personal assistant products, such as Apple Siri, Google Now, and Amazon Alexa, emerged in the last decades [5]. However, almost all existing AI personal assistants act as service terminals, who answer users’ questions, control home automation devices, play music, and even manage users’ calendars with verbal commands [6]–[9], [11].

There is a large number of routine dialogues in everyday life in which people need to deliver their requests to some staff service for hotel booking, movie ticket purchasing and restaurant reservation etc. Building AI personal assistants for such scenarios requires a role exchange from the service role to the requester roll, i.e. AI personal assistants are supposed to chat with real people on behalf of their users. The ultimate goal of such conversations is to deliver precise information efficiently according to the user request.

The existing conversation approaches [12], [14] usually employ encoder-decoder networks to generate machine responses directly from conversation context without explicit conversation policies. When playing a service roll, the under-lying conversation policies always converge to make sure that the important slots are filled. Such policies are user independent so that can be easily implicitly learned from conversations. When playing a requester roll, the expected policies aim to deliver specific user request precisely without missing anything. Implicit policy learning won’t reach this goal as the request fulfillment is not an objective in the existing approaches. On the other hand, to make the conversation
FIGURE 1. An example of Table-to-Dialogue generation. The personal assistant (PA) chats with the staff service (SS) based on the dialogue context and the request table. The latest PA response covers multiple slots from the table (the underline words).

efficient, machine generated responses are expected to carry multiple information slots, either requesting or delivering. This raises a new challenge for the response generation, i.e. how to generate semantic coherent machine response to carry multiple information slots.

In this paper, we present Table-to-Dialogue, a novel end-to-end approach to build AI personal assistants that chat with human staff services on behalf of their users. Each user request consists of a set of information slots that stored in a request table. As it is required to deliver user specific request precisely, the request table together with the conversation context is fed into a policy learning network, which learns explicit policies that describe what slots to address in the next round conversation. Since the policy output is a sequence of slots, the response generation needs to handle keywords from multiple slots. We further integrate a novel Multi-Slot Constrained Bi-directional Decoder (MS-CBD) into the above policy model, which generates machine response considering bi-directional context of the response.

We conduct empirical studies on the proposed approach to show the effectiveness against the state-of-the-art methods.

II. PROBLEM FORMULATION

This work aims to model a personal assistant by using a novel end-to-end Table-to-Dialogue generation method, where the personal assistant chats with staff services to deliver user request precisely. In this section, we give a brief problem definition of our task and describe the motivation of the proposed MS-CBD model.

A. PROBLEM DEFINITION

As shown in Figure 1, a user task is presented with a request table $T$, which is designed into hierarchical structure. The Context in Figure 1 is a conversation between a personal assistant (PA) and a staff service (SS). Each slot of the request table $T$ is composed of four components: Domain, Key, Intent and Value. The Domain denotes the domain information of the current dialogue, such as the restaurant domain in Figure 1. The Intent consists of three types: Inform, Request and Fail, where Inform and Request represents requesting or delivering information. Fail stands for that the personal assistant should inform another solution when the current request can not be satisfied. Given the context of a dialogue $X = (x_1, \ldots, x_k)$ and a request table $T$, our task is to generate a response sequence represented as $p(y|X, T)$, where $y$ denotes the generated response. The response should cover multiple slots $S = (s_1, \ldots, s_m)$ precisely from the request table, as shown in PA Response of Figure 1.
policy from the request table, a explicit policy learning is needed to represent the request information precisely. The policy learning is simplified to multi-slot generation in this paper. After learning the policy, a machine response generated according to the policy, which is expected to carry multiple information slots.

In addition, we observe the phenomenon that people generate a sentence from multi-keywords by keyword association ability and bi-directional thinking ability. People firstly associate several related word clusters for each keyword. Then they generate response leveraging the bi-directional context from the word clusters.

Motivated by this phenomenon, we decompose the mult-slot response generation into two steps. To simulate the association ability, we firstly use the intermediate representation of slot $s_i$ and $s_{i+1}$ to generate the word cluster $clu_i$ for the $i$-th segment. Then, to simulate the bi-directional thinking ability, we generate sentence $y$ by bi-directional attention on the overall word clusters to enhance the semantic coherence. The whole process can be formulated as the following equations:

$$p(y(T, X)) = p(y|S, T, X)p(S|T, X)$$
$$p(y|S, T, X) = p(y|clu, X)p(clu|S, X)$$

where $p(S|T, X)$ denotes the policy learning phase and $p(y|S, T, X)$ denotes the response generation phase. In the response generation phase, $p(clu|S, X)$ denotes the word cluster association and $p(y|clu, X)$ denotes the final response generation. The details of each phase are introduced in Section III.

III. APPROACH

In this section, we present our proposed model in detail. The model structure is illustrated in Figure 2, which consists of an encoder and a policy decoder integrated with a multi-slot constrained bi-directional decoder (MS-CBD). The encoder generate the representations of dialogue context and a request table. The MS-CBD generates the machine response carrying multi-slots according to the multiple slot values and their intermediate representations given by the policy decoder. In the following sub-sections, we will introduce each part of our model structure.

A. ENCODER

The encoder is composed of two parts: context encoder and table encoder, which encodes the dialogue context and a request table respectively.

1) CONTEXT ENCODER

The context encoder encodes the dialogue context into vector representation. We obtain each input word representation by a pre-trained word embedding. As shown in Context Encoder of Figure 2, the encoder runs a bi-directional long short-term memory network (BI-LSTM) [20] over the word representation, which encodes the input $n$ tokens into context representations $H \in \mathbb{R}^{n \times d_m}$, where $d_m$ is the dimension of encoder.

2) TABLE ENCODER

Given the request table $T$, the table encoder aims to generate the context-aware slot representation from the context and table, As shown in Figure 2. The intrinsic of slot embedding $E \in \mathbb{R}^{m \times d_m}$ is generated by a linear layer after concatenating constituent component embeddings, where $m$ denotes the number of slots in table $T$.

The table encoder further performs context-aware attention to capture the interaction of dialogue context and the request table. The attention matrix $\hat{E}$ is calculated according the following equation:

$$\hat{E} = \text{softmax}(EW_t^T)H$$

where the shape of $\hat{E}$ is same with the shape of $E$ and $W_t \in \mathbb{R}^{d_t \times d_m}$ is trainable parameter. We finally concatenate the slot embedding and attention to generate the context-aware slot representation $S = [\hat{E}; E]$.

B. DECODER

To generate the response covered multiple information slots, a policy decoder is used to learn conversation policy. The policy here denotes the inferred slot sequences from the request table. Further more, a novel multi-slot constrained bi-directional decoder (MS-CBD), which is integrated into the decoder network, is used to generate the response according to the multiple slot values and their intermediate representations from the policy decoder.

1) POLICY DECODER

The policy decoder uses the RNNs to model the generation of a slot sequence from the request table. The hidden state at the
i-step $d_i$ is computed by a LSTM function, which inputs are previously predicted hidden state $d_{i−1}$ and slot representation $s_{i−1} \in S$. We use $D \in \mathbb{R}^{l \times d_h}$ to denote the policy decoder hidden matrix, where the $l$ denotes the length of slot sequence and $d_h$ is the hidden size.

The probability matrix $P_s \in \mathbb{R}^{l \times m}$ of slot generation is calculated via a pointer network [21], where $l$ denotes the length of slot sequence and $m$ denotes the number of slots in the request table, as shown in following equation:

$$P_s = \text{softmax}(DW_1S^T)$$
$$\hat{D} = \tanh(C_s; D[W_2])$$
$$C_s = \text{softmax}(DW_2H^TH)H$$

(4)

where $C_s$ denotes the context attention matrix, whose shape is the same as $D$. To generate the output $D$ of the policy decoder, a non-linear layer is conducted on the concatenation of the decoder hidden matrix $D$ and the attention matrix $C_s$. $W_1 \in \mathbb{R}^{d_u \times d_m}$, $W_2 \in \mathbb{R}^{2d_u \times d_h}$ are the trainable network parameter. We finally obtain a slot sequence $S = (s_1, s_2, \ldots, s_m)$ from the policy decoder.

2) RESPONSE DECODER

The MS-CBD generates segment between each adjacent slots in a keyword association and bi-directional thinking ability way, as illustrated in Figure 2. To simulate the keyword association ability, the intermediate representation of policy decoder is used to generate the word cluster $clu$ of each segment. Then, to simulate the bi-directional thinking ability, the response decoder performs a bi-directional attention over overall word clusters when generating response.

The first step is to associate word cluster using the intermediate representation of the policy decoder. We concatenate the hidden states of two adjacency slots to represent segment state:

$$g_i = [d_i; d_{i+1}]$$

(5)

where $i$ denotes the $i$-th segment of the sequence, and $d_i \in D$. Then, the segment state is used to generate the word cluster $clu_i$. We calculate the possibility of selecting $w_d$ in the $i$-th segment via:

$$p(w = w_d|s_i, s_{i+1}, X) \propto \exp(e_{ij}^T W_r g_i)$$

(6)

where $e_{ij}$ are the embedding vector for word $w_d$ and trainable parameter $W_r \in \mathbb{R}^{d_u \times 2d_h}$. For the $i$-th segment, we use top-k relevant words as cluster $clu_i$ to aggregate the distribution of words in the segment.

Then, the response decoder generates tokens between two adjacent slots by RNNs in a bi-directional thinking way. The decoder leverage bi-directional attention of the overall word clusters $clu$ when generating current token to enhance the semantic coherence of generated response. To make use of the position between each word cluster, inspired by relative position representations [22], we further incorporate the relative position vector into word embedding in cluster. Specifically, when decoding the tokens in the $i$-th segment, the representation for the $k$-th word in $j$-th cluster $clu_j$ is computer as:

$$\hat{e}_k = e_k + p_{j-i}$$

(7)

where $e_k$ is the embedding for the $k$-th word and the $p_{j-i}$ denotes the trainable relative position vector. Then we calculate bi-directional attention score between decoder hidden state $v_t$ at time step $t$ with word $w_d$ via a dot-similarity $\odot()$ and get the cluster-aware representation $o_t$, similar to Equation 3. Finally, the response decoder maintains generated state by delivering it in a left-to-right manner among segments via RNNs and calculates the possibility of generated word $w_d$ via:

$$p(y = w_d|clu, X) \propto \exp(e_d^T \hat{v}_t)$$
$$\hat{v}_t = \tanh(W_v[v_t; c_i; o_t])$$

(8)

where $c_i$ is the context attention vector the same as Equation 4 and $W_v \in \mathbb{R}^{(2d_u+t_d) \times d_h}$ is trainable parameter.

Taking Figure 2 as an example, the decoder generates the yellow tokens from the slot ‘BEG’ until the slot ‘Italy’ in purple and passes the hidden states to the next segment.

IV. EXPERIMENT

In this section, we first show the policy and response generation results of MS-CBD compared with state-of-the-art conversation approaches. Then, we evaluate the personal assistant in human rating manner, and conduct case visualization.

A. EXPERIMENT SETUP

1) DATASET

We conduct our experiments on the MultiWOZ [14], the latest human-to-human multi-turn dataset for Task-Oriented dialogue across several domains, which contains 10,438 dialogues on 6 different domains. To build the Table-to-Dialogue task where personal assistant servers as a request role, we extract the request tables from user’s dialogue act in conversation and split the conversation into fixed context with labeled policy and response in a single round. In total, there are 30,275/4,071/4,083 samples for training, validation and testing, respectively. Specially, 59.9% of dialogues contain more than 6 slots in the request table and the average slot sequence length for a response is 1.24.

2) BASELINES

We compare our proposed model against the following approaches:

- **Seq2Seq**: a seq2seq model with attention mechanism [13] generates response from left to right. We also evaluate the CopyNet [23] that the seq2seq model augmented with a copying mechanism. To compare fairly, we integrate baseline models with our table encoder.

- **MD-DAS**: a multi-domain dialogue architecture [14] for Context-to-Text generation encodes the policy implicitly and generates response from the hidden representation and context.
• **FGSD**: a field-gating seq2seq with dual attention [24] proposed for the Table-to-Text task and achieves state-of-the-art result on the WikiBio [32]. We encode the request table with field-gating encoder and apply the dual attention between key and value component of slot in the decoder.

• **ABDNMT**: a sequence generation model uses backward decoder and forward decoder to explore bi-directional information for neural machine translation [35]. We integrate the model with our policy decoder to predict slot explicitly for comparing fairly with our response decoder.

3) METRICS

We use three kinds of metric to evaluate the Table-to-Dialogue task. (1) **NLG Metrics**. Following original settings [14], we measure fluency with BLEU score [26]. Considering the lack of reference labels, we further use NIST, METEOR and ROUGE [27]–[29] to evaluate the similarity between generated output and labeled response. (2) **Policy Metrics**. We compute F1 and Acc scores between predicted and labeled slot sequence. The F1 score measures slots overlap while the Acc measures the predicted sequence as a whole is equivalent to the label. (3) **Human Rating**. The human evaluator is asked to label the conversation with 2-level score, 0 and 1. If our system fulfills the request table (including duplication) without mistaking the request, the human evaluator will label it as 1; otherwise, the label is 0. We use Acc to measure the result.

4) IMPLEMENTATIONS

The dimensions of word embedding, attention matrix and encoder are set to 300. The hidden size of slot and response decoder are set to 100. Word embedding is initialized with Glove [33] and shared between the context and table encoder. They are fixed during training. We train the slot decoder and response decoder with teacher forcing learning [31] by words between two adjacent slots (including slot itself) as labels by the bag-of-word loss [30]. In our experiment, We use top-5 most related words as word cluster. We optimize models using Adam [36] with default hyperparameters and the batch size is set to 64.

**B. POLICY RESULTS**

To conduct a detailed analysis of policy generation, we evaluate the result from **component and sequence** levels. We measure the F1 score on **Domain, Intent and Key** between prediction and ground true sequence on **component** level. To measure the output sequence as a whole on **sequence** level, we use Acc metric. Table 1 presents the results of MS-CBD on various baselines on the test set. MS-CBD clearly outperforms other baselines in both **component and sequence** levels. Performing context-aware attention (CA) brings about 1.4% and 1.0% improvement in **Intent** and **Key** components. In contrast to learning conversation policy implicitly (Seq2Seq, CopyNet, MD-DAS, FGSD), the model learning explicitly (ABD-NMT, MS-CBD) exhibits much better performance in policy generation, especially in the **Key** component. When we replace the MS-CBD with a vanilla LSTM decoder, which generates response with implicit policy learning, the performance drop dramatically about 20.3% in the **sequence** level. These results testify our hypothesis that a explicit policy is essential in delivering specific user request precisely.

**C. SEQUENCE GENERATION RESULTS**

We first investigate the effectiveness on response generation in a single round. As shown in Table 2, MS-CBD significantly outperforms all the baselines by a substantial margin on all the **NLG metrics**. It obtains 3.7% BLEU score absolute improvement over the state-of-the-art model FGSD [24] of Table-to-Text generation. To study the performance of multi-slot constrained generation, we use the ground true labeled slot sequence as an input, which is depicted in Table 2. We find that MS-CBD still achieves the best performance among various baselines. Moreover, we draw the following conclusions:

1) The result of our model with **Associate** performs better (+2.6% BLEU), which verifies our speculation that word clusters can simulate the human association ability, and therefore, the simulation will lead to a better generation. We further explore the effectiveness of the number of words in the cluster, as shown in Figure 4. We observe that the performance of response generation and word association both draw back when the cluster contains more than 7 words. This phenomenon, on the one hand, demonstrates the association ability of cluster when containing proper number of words. On the other hand, it shows the word cluster has up bound in the improvement. One underlying reason may be that the feature redundancy confuse the decoder in selecting related words.

2) When using the bi-directional attention over word clusters, the BLEU score increases 2.2% and 1.9% in response and multi-slot constrained generation, respectively. Compared with ABDNMT [35], which explores bi-directional context with a backward decoder and integrated our explicit policy decoder, it still has 1.2% BLEU score absolute improvement in response generation. This demonstrates that

**TABLE 1.** The results of policy generation in component and sequence (Seq) level. We evaluate the components of slot using F1 metric and evaluate the sequence as a whole using Acc. -CA means removing context-aware attention and -Dec means replacing the decoder with a vanilla LSTM decoder.

| Method      | Domain | Component | Intent | Key | All  | Seq | Acc |
|-------------|--------|-----------|--------|-----|------|-----|-----|
| Seq2Seq     | 80.3%  | 71.9%     | 37.2%  | 31.4% | 31.4% | 11.1% |
| CopyNet     | 74.4%  | 70.0%     | 44.4%  | 41.1% | 41.1% | 18.8% |
| MD-DAS      | 79.5%  | 71.3%     | 66.2%  | 58.4% | 58.4% | 31.4% |
| FGSD        | 94.2%  | 85.4%     | 68.9%  | 66.9% | 66.9% | 39.4% |
| ABDNMT      | 94.1%  | 86.9%     | 72.1%  | 70.4% | 70.4% | 41.5% |
| MS-CBD      | 94.4%  | 88.2%     | 74.4%  | 72.5% | 72.5% | 42.5% |
| -CA         | 94.3%  | 86.8%     | 73.4%  | 72.4% | 72.4% | 42.0% |
| -Dec        | 71.6%  | 68.1%     | 43.4%  | 42.6% | 42.6% | 22.2% |
the bi-direction attention mechanism leverages richer coherence over the generated response. In addition, the relative position representation models the positional information of word clusters, which improves the BLEU score nearly 1.3% in response generation.

(3) Particularly, MS-CBD performs better when generated response contains more slots. As shown in Figure 3, the increment of BLEU score between two response groups of MS-CBD increases when containing more slots, while other baselines decline between the groups containing 4 and 3 slots. The result demonstrates MS-CBD is more adaptive to other baselines in multi-slot constrained generation.

D. END-TO-END CONVERSATION RESULTS

Except for the policy and generation results, we also conduct a end-to-end conversation experiment. We let three human evaluators, who serve as the customer service staff, to chat with our proposed system. Each human evaluator complement 30 conversation using Human Rating. Each conversation has a corresponding request table in advance. The end-to-end conversation result is shown in Table 3. Compared to other conversation approaches, our proposed model outperforms in delivering user request precisely with end-to-end dialogue manner. Our method has nearly 4.2% improvement compared to MD-DAS, which encodes the conversation policies implicitly. We also find that our proposed model achieves the task fulfillment within fewer turns, which demonstrates the efficiency of our model under multi-slot generation.

E. VISUALIZATION OF MS-CBD

To better validate MS-CBD is able to associate related words with keyword and select informative clusters, we visualize the attention score between generated tokens and word clusters using Vig [34] in Figure 5.

Figure 5 illustrates the word cluster and bi-direction thinking mechanism. Every dotted box is the cluster associated by the slot in pink box. Blue line denotes attention score.

TABLE 2. Sequence generation results for our MS-CBD and other baselines on the MultiWOZ test set in a single round. We use various NLG metrics to evaluate the generation result. The Response presents the results based on the given dialogue context and a request table, while the Multi-slot Constrained denotes the results based on feeding the model with the ground true slot sequence.

| Approach | Response BLEU | NIST | METEOR | ROUGE | Multi-Slot Constrained BLEU | NIST | METEOR | ROUGE |
|----------|---------------|------|--------|-------|-----------------------------|------|--------|-------|
| Seq2Seq  | 10.5%         | 2.28 | 14.7%  | 32.0% | 17.5%                       | 3.59 | 21.4%  | 43.0% |
| CopyNet  | 10.5%         | 2.33 | 14.3%  | 32.0% | 17.5%                       | 3.65 | 21.4%  | 43.2% |
| FGSD     | 12.2%         | 2.83 | 15.4%  | 33.1% | 18.8%                       | 4.19 | 22.3%  | 43.2% |
| MD-DAS   | 13.5%         | 3.13 | 16.7%  | 35.2% | 18.6%                       | 4.15 | 22.1%  | 44.1% |
| ABDNMT   | 14.7%         | 3.31 | 17.9%  | 37.1% | 20.2%                       | 4.23 | 23.1%  | 45.4% |
| MS-CBD   | 15.9%         | 3.60 | 19.1%  | 38.9% | 22.1%                       | 4.56 | 23.4%  | 47.7% |

FIGURE 3. The increment of BLEU score between response groups divided according to the slot sequence length. For example, “1->2” denotes the increment between two response groups which the length of slot sequence is 2 and 1.

FIGURE 4. The experiment result of using different top-k words to represent the word cluster. The left axis denotes the F1 score between word cluster and words in ground true slot segment. The right axis denotes the BLEU score.

TABLE 3. End-to-end conversation results on the MultiWOZ testset by human rating. We evaluate the result by the Acc metric. We also report the average number of turns per dialogue ending with Human Rating 1.

| Approach | Number of Turns | Acc   |
|----------|-----------------|-------|
| CopyNet  | 11.7            | 7.3%  |
| MD-DAS   | 11.4            | 8.2%  |
| FGSD     | 10.8            | 10.6% |
| ABDNMT   | 10.3            | 11.0% |
| MS-CBD   | 9.8             | 12.4% |

FIGURE 5. The word cluster and bi-direction thinking mechanism. Every dotted box is the cluster associated by the slot in pink box. Blue line denotes attention score.
In example (a), we observe that the attention score between ‘leaving’ and ‘after’ is evidently higher than others, suggesting that the bi-directional decoder has learned which word clusters are more informative. The association ability simulated by word cluster also enhances the generation performance. For example, the slot ‘moderate’ of second cluster in example (c) associates the word ‘price’, which has a strong impact in the next step generation. The visualization show the effectiveness of the MS-CBD.

V. RELATED WORK

A. AI PERSONAL ASSISTANT MODELING

Research on AI personal assistants has been studied for decades from both academia and industry [1]–[3]. Most existing AI personal assistants perform their tasks by acting as service terminals [9]. For example, [6] proposes a assistant agent to manage users’ calendars, while Cmarad Azaria [8] proposes a programmable personal assistants for detecting new emails, playing music, etc. Matsuyama et al. [9] demonstrates a socially-aware assistant to analysis user’s behaviours. Although these service terminals assistance achieve great success in the last decades [5], building AI personal assistants as a requester roll, however, still needs more attention to be explored. In this paper, we focus on building personal assistants on behalf of user, where they chat with people according to pre-defined user request tables.

B. TABLE-TO-TEXT

The insight behind performing Table-to-Dialogue approach to build personal assistant is partly inspired by the success of text generation from a factual table [10], [14], [15], [32]. Similar to Table-to-text generation task [24], Table-to-Dialogue also requires model to understand the context of given table. In the context of dialogue, Budzianowski et al. [14] proposes a text generation task, which requires to generate response from conversation and oracle belief-state to fulfill certain tasks as a service roll. Different from the previous studies, to server as a requester role in our task, the expected response generation should deliver specific user request precisely, raising new challenge on dialogue generation.

C. TEXT GENERATION

Another research area related to our work is the text generation [16]–[18], [35]. Most works are focus on the generation without keywords constraint [16], [23], [35]. When it comes to multi-keyword constrained generation, it brings a new challenge for generation. Uchimoto et al. [17] first describes a system for generating text from multi-keyword by generation rules. Chen et al. [19] proposed a hierarchical neural network to generate response from the implicit semantically condition. To the best of our knowledge, MS-CBD first performs multi-keyword generation using keyword association and bi-directional context in a human manner.

VI. CONCLUSION

In this paper we try to build AI personal assistants for a different yet important dialogue scenario, where they chat with people to fulfill specific tasks on behalf of their human users. We present Table-to-Dialogue to address the challenges when building a requester roll AI personal assistant. In Table-to-Dialogue, explicit conversation policy are learned to decide the information slots in the next round of the dialogue based on the conversation context and the user request table. We also integrate a novel MS-CBD model into the policy network, to generate machine response according to the multiple slot values and their intermediate representations from the policy decoder. MS-CBD leverages the bi-directional context of the response when generating it to enhance the semantic coherence. The experiments shows that our approach significantly outperform the state-of-the-art conversation approaches.

REFERENCES

[1] S. Levinson and K. Shipley, “A conversational-mode airline information and reservation system using speech input and output,” in Proc. ICASSP, 1980, pp. 119–137.
[2] S. Janarthanam, O. Lemon, X. Liu, P. Bartie, W. Mackaness, and T. Dalmas, “A multithreaded conversational interface for pedestrian navigation and question answering,” in Proc. SIGDIAL, 2013, pp. 151–153.
[3] R. Guha, V. Gupta, V. Raghunathan, and R. Srikant, “User modeling for a personal assistant,” in Proc. WSDM, 2015, pp. 275–284.

[4] I. Labutov, S. Srivastava, and T. Mitchell, “LIA: A natural language programmatic personal assistant,” in Proc. Syst. Demonstrations (EMNLP), Nov. 2018, pp. 145–150.

[5] V. Krumska and G. Bobahuta, “Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google home),” in Proc. CCWC, 2018, pp. 99–103.

[6] P. J. Modi, M. Veloso, F. S. Smith, and J. Oh, “CMRadar: A personal assistant for calendar management,” in Proc. NACL, 2004, pp. 169–181.

[7] M. Freed, J. G. Carbonell, G. J. Gordon, J. Hayes, B. A. Myers, D. P. Skowronik, S. F. Smith, A. Steinfeld, and A. Tomicak, “RADAR: A personal assistant that learns to reduce email overload,” in Proc. AAAI, 2008, pp. 1287–1293.

[8] A. Azarja, J. Krishnamurthy, and T. Mitchell, “Instructable intelligent personal agent,” in Proc. AAAI, 2016, pp. 2681–2689.

[9] Y. Matsuyama, A. Bhardwaj, R. Zhao, O. Romeo, S. Akoju, and J. Cassell, “Socially-aware animated intelligent personal assistant agent,” in Proc. SIGSLID, Sep. 2016, pp. 224–227.

[11] J. Yin, X. Jiang, Z. Lu, L. Shang, H. Li, and X. Li, “Neural generative question answering,” in Proc. NAACL Workshop HCQA, 2016, pp. 36–42.

[12] I. Labutov, S. Srivastava, and T. Mitchell, “LIA: A natural language programmatic personal assistant,” in Proc. EMNLP-System Demonstrations, Nov. 2018, pp. 145–150.

[13] T. Zhao, K. Xie, and M. Eskofnari, “Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models,” in Proc. NAACL, 2019, pp. 1208–1218.

[14] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in Proc. ICLR, 2015, pp. 1–15.

[15] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. N. Casanueva, S. Ultes, O. Ramadian, and M. Gašić, “MultiWOZ—A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling,” in Proc. EMNLP-System Demonstrations, Nov. 2018, pp. 5016–5026.

[16] J. Bao, D. Tang, N. Duan, Z. Yan, M. Zhou, and T. Zhao, “Text generation from tables,” IEEE/ACM Trans. Audio, Speech, Language Process., vol. 27, no. 2, pp. 311–320, Feb. 2019.

[17] X. Geng, X. Feng, B. Qin, and T. Liu, “Adaptive multi-pass decoder for neural machine translation,” in Proc. EMNLP, 2018, pp. 523–532.

[18] K. Uchimoto, S. Sekine, and H. Ishahara, “Text generation from keywords,” in Proc.icoling, 2002, pp. 1–7.

[19] Z. Song, L. Liu, W. Song, X. Zhao, and C. Du, “A neural network model for Chinese sentence generation with key word,” in Proc. iccvec, 2019, pp. 334–337.

[20] W. Chen, J. Chen, P. Qin, X. Yan, and W. Y. Wang, “Semantically conditioned dialog response generation via hierarchical disentangled self-attention,” in Proc. ACL, 2019, pp. 3696–3709.

[21] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997.

[22] O. Vinyals, M. Fortunato, and N. Jaitly, “Pointer networks,” in Proc. NIPS, 2015, pp. 2692–2700.

[23] P. Shaw, J. Uszkoreit, and A. Vaswani. “Self-attention with relative position representations,” in Proc. NAACL, 2018, pp. 464–468.

[24] J. Gu, Z. Lu, H. Li, and Q. K. V. Li, “Incorporating copying mechanism in sequence-to-sequence learning,” in Proc. ACL, 2016, pp. 1631–1640.

[25] T. Liu, K. Wang, L. Sha, B. Chang, and Z. Sui, “Table-to-text generation by structure-aware seq2seq learning,” in Proc. AAAI, 2018, pp. 4881–4888.

[26] X. Zhang, J. Su, Y. Qin, Y. Liu, R. Ji, and H. Wang, “Asynchronous bidirectional decoding for neural machine translation,” in Proc. AAAI, 2018, pp. 5698–5705.

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “BLEU: A method for automatic evaluation of machine translation,” in Proc. ACL, 2002, pp. 311–318.

[28] G. Doddington, “Automatic evaluation of machine translation quality using N-gram co-occurrence statistics,” in Proc. ICHTIR, 2002, pp. 138–145.

[29] C.-Y. Lin, “ROUGE: A package for automatic evaluation of summaries,” in Proc. ACL Workshop Text Summarization Branches Out, Jul. 2004, pp. 74–81.

[30] S. Banerjee and A. Lavie, “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments,” in Proc. ACL Workshop Intrinsic Extrinsic Eval. Measures Mach. Transl. Summar.ization, Jun. 2005, pp. 65–72.

[31] T. Zhao, R. Zhao, and M. Eskenazi, “Learning discourse-level diversity for neural dialog models using conditional variational autoencoders,” in Proc. ACL, 2017, pp. 654–664.

[32] J. Ronald Williams and D. Zipser, “A learning algorithm for continually running fully recurrent neural networks,” Neural Comput., vol. 1, no. 2, pp. 270–280, 1989.

[33] R. Lebret, D. Grangier, and M. Auli, “Neural text generation from structured data with application to the biography domain,” in Proc. EMNLP, 2016, pp. 1203–1213.

[34] J. Pennington, R. Socher, and C. Manning, “GloVe: Global vectors for word representation,” in Proc. EMNLP, 2014, pp. 1532–1543.

[35] J. Vig, “A multiscale visualization of attention in the transformer model,” in Proc. ACL Syst. Demonstrations, Jul. 2019, pp. 37–42.

[36] X. Zhang, J. Su, Y. Qin, Y. Liu, R. Ji, and H. Wang, “Asynchronous bidirectional decoding for neural machine translation,” in Proc. AAAI, 2018, pp. 5698–5705.

[37] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in Proc. ICLR, 2014, pp. 1–15.