Phrase-Level Action Reinforcement Learning for Neural Dialog Response Generation

Takato Yamazaki
The University of Tokyo, Japan
takato@yamazaki.dev

Akiko Aizawa
The University of Tokyo, Japan
National Institute of Informatics, Japan
aizawa@nii.ac.jp

Abstract

Defining a sophisticated action space for a dialog agent is essential for efficient training with reinforcement learning (RL). Recent work introduces discrete latent variables to use as an action space; however, a limitation is that a global vector can contain entangled information such as dialog act, sentence structure, and content. This sacrifices the flexibility of the response generation. In this paper, we propose phrase-level action reinforcement learning (PHRASERL), which allows the model to flexibly alter the sentence structure and content with the sequential action selection. Our model first learns to generate useful phrases during the supervised pre-training, and then further trained to form a response by rearranging the phrases with reinforcement learning. Experiments on the MultiWOZ dataset show that our model achieves competitive results with state-of-the-art models on automatic evaluation metrics, indicating that our phrase-level action space has improved flexibility and is effective for solving task-oriented dialogs.

1 Introduction

Dialog policy optimization is key research to efficiently solving real-world tasks (Rastogi et al., 2020; Budzianowski et al., 2018; Lewis et al., 2017). In neural response generation, which has made remarkable progress in recent years (Vinyals and Le, 2015; Li et al., 2016a; Serban et al., 2017; Bao et al., 2019), many methods that apply reinforcement learning (RL) have been proposed (Li et al., 2016b; Peng et al., 2018; Saleh et al., 2019; Zhao et al., 2019). In those studies, one major issue was how to define an action space. Early research proposed a method in which each word of the response is an action (Li et al., 2016b). However, this has a shortcoming that the generated responses deviate from natural human language (Zhao et al., 2019). A possible reason is that the action space is huge, making it difficult to optimize with RL. Moreover, rewarding only the task accomplishment can cause biased improvement, which leads the model to ignore the comprehensibility of the generated response (Wang et al., 2020a).

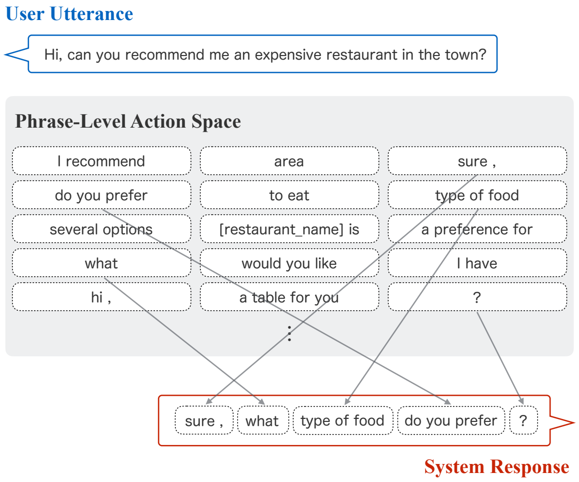

Figure 1: Demonstration of phrase-level action. The model generates useful phrases and rearrange them to form a response.

To overcome such issues, LaRL (Zhao et al., 2019) was proposed, which used a discrete global vector to represent dialog acts. In this method, reinforcement learning is performed only on those discrete latent variables, thus the policy optimization is achieved without affecting the language generation. However, LaRL depends on a single vector from the beginning to the end during the response generation, even though a response may often contain more than one dialog act and contents (Wang et al., 2020b). Due to this, a static, global vector tends to be an entangled representation of multiple dialog acts, sentence structure, and contents. Therefore, using a global vector for action space sacrifices the flexibility of the response generation.

To improve the flexibility of the surface realiza-

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 5028–5038
August 1–6, 2021. ©2021 Association for Computational Linguistics

tion, we propose phrase-level action reinforcement learning (PHRASERL), in which the model performs action selections in fine-grained semantic units. PHRASERL is based on neural hidden semi-Markov model (HSMM) decoder (Wiseman et al., 2018) which generates typed text-segments from hidden states, and we use them as an action space. This disentangles the generation process: the policy learns to structure a response as a sequence of hidden states, while each hidden state is trained to represent content or a type of phrase. Intuitively, as described in figure 1, our model learns to generate useful phrases during the supervised pre-training, and it is further trained with reinforcement learning to reorder the phrases and form a response.

Experiment results on the task-oriented MultiWOZ dataset (Budzianowski et al., 2018) show that our best performing model outperforms LaRL by far and achieves competitive results with the state-of-the-art models in automatic evaluation. Furthermore, PHRASERL can maintain a high BLEU score, suggesting that the model is flexible in its output response depending on the context. Finally, we study the phrase generation from hidden states in a case study, and show that the hidden state-action space is capable of generating (1) informative response, (2) grammatical sentence, and (3) diverse intentions, which can be considered as requirements for an effective action space. Our code is available at https://github.com/Alab-NII/PhraseRL.

2 Related Work

A classical approach for realizing task-oriented dialog systems is the frame-based dialog system (Chen et al., 2017). This model generates a response in a pipeline fashion, by splitting the generation process into three modules: natural language understanding, dialog management, and natural language generation. Natural language understanding converts user utterances to a semantic frame which is considered a dialog state, and a popular method is slot filling (Mrkšić et al., 2017; Ramadan et al., 2018). The estimated dialog state is then passed on to dialog management to determine the next action, which is formulated as a partially-observable Markov decision process (POMDP) (Young, 2006). The action space is represented with hand-crafted dialog acts (Budzianowski et al., 2018; Stolcke et al., 2000) or meaning representations (Balakrishnan et al., 2019). Finally, a natural language generator generates a response, which is often realized with recurrent neural networks (Zhou et al., 2016; Tran and Nguyen, 2017). Our proposed model spans between dialog management and natural language generation, however, our model does not require any hand-crafted representation.

Past works that applied reinforcement learning to dialog models have shown a huge performance improvement in task success (Lewis et al., 2017; He et al., 2018). Li et al. (2016b) proposed a dialog generation method by using deep reinforcement learning with words as action spaces. Although the rewards were carefully designed, it is reported that these models tend to generate incomprehensible responses. Zhao et al. (2019) solved the problem by using discrete latent variables as the action space. Wang et al. (2020a) have extended LaRL and applied hierarchical reinforcement learning technique to decouple the dialog policy and natural language generation. The model is composed of two policy networks; one is the high-level policy which acts on latent dialog act and another is the low-level policy that acts on words. The low-level policy is prone to degeneration, so the paper proposes to use language model discriminator as a reward provider. These models either use words or a global latent variable as an action space, however, our work stands between the two by using phrases for the action space.

3 Preliminaries

In this section, we first explain the characteristics and formulation of the HSMM. We then describe the neural HSMM decoder, which will be the backbone of our proposed method.

3.1 Hidden Semi-Markov Model (HSMM)

In our work, we consider sentences as a sequence of phrases, and the probabilistic model that can represent this is the hidden semi-Markov model (HSMM). The difference between a standard hidden Markov model (HMM) and HSMM is shown in figure 2. While an HMM gives one observation from a hidden state, an HSMM gives a sequence of observations per hidden state. Therefore, if we consider words as observations, hidden states will be phrases.

To represent a variety of sentences with a limited number of hidden states, HSMM is expected to assign the same sequence of hidden states for similar sentences. Figure 3 is an example. As it can

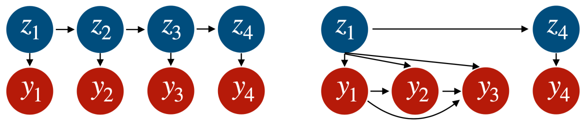

Figure 2: The difference between Hidden Markov Models (left) and Hidden Semi-Markov Models (right)

[what]$_{8}$ [time]$_{23}$ [would you like]$_{11}$ [to leave ?]$_{34}$
[what]$_{8}$ [kind of food]$_{23}$ [would you like]$_{11}$ [to eat ?]$_{34}$

Figure 3: An example of sentence segmentation in HSMM. The number represents the index of a hidden state.

be seen, each hidden state has a type; for instance, hidden state #8 and #11 outputs the same phrase, while #23 outputs noun phrases and #34 outputs verb phrases for end of questions. In this way, text-segments assigned to a certain hidden state will be having a similar property.

For our model, we specifically use *conditional* HSMMs which takes a source input x. For each timestep $t \in \{1, \cdots, T\}$, we denote the observations as $y_1 \cdots y_T$ and the discrete hidden states as $z_t \in \{1, \cdots, K\}$. We additionally introduce two latent variables; the length of the current observation sequence, denoted as $l_t \in \{1, 2, \cdots, L\}$, and a binary variable which represents whether the sequence is finished at timestep t, denoted as f_t. The maximum number of hidden states K and observation length L are tunable parameters. An HSMM will be represented with a joint distribution of the observations and the described latent variables:

$$\begin{aligned} p(y, z, l, f|x; \theta) = & \prod_{t=0}^{T-1} p(z_{t+1}, |z_t, x)^{f_t} \\ & \times \prod_{t=0}^{T-1} p(l_{t+1}|z_{t+1})^{f_t} \\ & \times \prod_{t=1}^{T} p(y_{t-l_t+1:t}|z_t, l_t, x)^{f_t}. \end{aligned} \tag{1}$$

In other words, an HSMM will be the product of three probabilities: state transition distribution, length distribution, and emission distribution.

3.2 Neural HSMM Decoder

We now introduce a neural HSMM decoder (Wiseman et al., 2018). Figure 4 shows the overview of the decoder model. The aforementioned three distributions can be obtained using trainable parameters. We define the embeddings of the x as $\boldsymbol{x} \in \mathbb{R}^d$ and the hidden state z as $\boldsymbol{z} \in \mathbb{R}^d$.

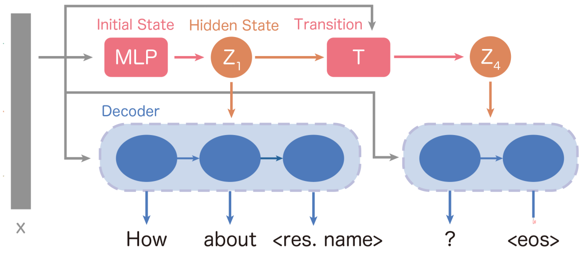

Figure 4: Model overview of neural HSMM decoder.

State Transition Distribution For the state transition function $p(z_{t+1}|z_t, x)$, we use $K \times K$ matrix, where sum of each row is 1. We define the state transition matrix as

$$p(z_{t+1}|z_t, x) \propto \boldsymbol{A}\boldsymbol{B} + \boldsymbol{C}(\boldsymbol{x})\boldsymbol{D}(\boldsymbol{x}), \tag{2}$$

where $\boldsymbol{A} \in \mathbb{R}^{K \times m_1}$ and $\boldsymbol{B} \in \mathbb{R}^{m_1 \times K}$ represents state embeddings, and where $\boldsymbol{C} : \mathbb{R}^d \to \mathbb{R}^{K \times m_2}$ and $\boldsymbol{D} : \mathbb{R}^d \to \mathbb{R}^{m_2 \times K}$ is a non-linear function parameterized with neural networks. m_1 and m_2 are tunable parameters.

Length Distribution Wiseman et al. (2018) have found that parameterizing length distribution leads to hidden states that specialize in specific output lengths. To avoid that, we simply used uniform distribution for every length probability $p(l_{t+1}|z_{t+1})$.

Emission Distribution For the emission distribution $p(y_{t-l_t+1:t}|z_t, l_t, x)$, we use the product of the token probability. Therefore, the emission distribution is obtained with

$$\begin{aligned} & p(y_{t-l_t+1:t}|z_t = k, l_t = l, x) = \\ & \prod_{i=1}^{l_t} p(y_{t-l_t+i}|y_{t-l_t+1:t-l_t+i-1}, z_t = k, x) \\ & \times p(\langle eop \rangle|y_{t-l_t+1:t}, z_t = k, x) \times \mathbf{1}_{\{l_t = l\}}, \end{aligned} \tag{3}$$

where $\langle eop \rangle$ stands for end-of-phrase token which indicates the end of emission for each hidden state. We use gated recurrent unit (GRU) to compute the token probabilities:

$$\boldsymbol{v}_i = \boldsymbol{W}\mathrm{ReLU}\left(\mathrm{GRU}\left(\left[\boldsymbol{y}'_{i-1}, \boldsymbol{z}^k\right], \boldsymbol{h}_{i-1}\right)\right), \tag{4}$$

where $\boldsymbol{y}'_{i-1}$ is the embeddings of the generated previous token and $\boldsymbol{z}^k$ is the embeddings of the hidden state k. Finally, the probability of the token w will be

$$p(y_{t-l_t+i} = w|z_t = k, l_t = l, x) = v_{i,w}. \tag{5}$$

Training We assume z, l, f of an HSMM is unobservable, so we maximize the marginal likelihood of the emission y given only input x by training. The marginal likelihood of y in HSMMs can be efficiently computed using a dynamic programming algorithm such as backward-algorithm (Murphy, 2002). Using variables β, β^*, the backward-algorithm can be expressed as

$$\begin{aligned}\beta_t(j) &= p(y_{t+1:T}|z_t = j, f_t = 1, c) \\ &= \sum_{k=1}^{K} \beta_t^*(k) p(z_{t+1} = k|z_t = j)\end{aligned} \quad (6)$$

$$\begin{aligned}\beta_t^*(k) &= p(y_{t+1:T}|z_{t+1} = k, f_t = 1, c) \\ &= \sum_{l=1}^{L} [\beta_{t+l}(k) p(l_{t+1} = l|z_{t+1} = k) \\ &\qquad p(y_{t+1:t+l}|z_{t+1} = k, l_{t+1} = l)],\end{aligned} \quad (7)$$

where $\beta_T(j) = 1$. Finally, from the definition $f_0 = 1$, the log-marginal likelihood of y will be:

$$\ln p(y|x; \theta) = \ln \sum_{k=1}^{K} \beta_0^*(k) p(z_1 = k). \quad (8)$$

Here, we compute $p(z_1 = k)$ with a linear layer. Since equations (6) and (7) are differentiable, we can optimize θ by maximizing the log-marginal likelihood $\ln p(y|x; \theta)$ with backpropagation.

4 Proposed Method

The original neural HSMM decoder (Wiseman et al., 2018) was proposed as a data-to-text generation method, so the model needs modification to be applied to dialog response generation. Particularly, we investigate what the HSMM should condition on, in other words, we determine the input x. However, we need to carefully design x because of a known problem of the neural HSMM decoder, which will be explained first. Afterward, we discuss how to improve the response quality by applying reinforcement learning.

4.1 Conditional Source for Neural HSMM Decoder

The neural HSMM decoder is based on the assumption that the output phrases from each hidden state to be independent of each other. However, if the source $\boldsymbol{x}$ is informative enough to capture the interdependence between the phrases, the RNN decoder may fully depend on the source $\boldsymbol{x}$ and ignore the hidden state $\boldsymbol{z}$ for the generation. We will call this problem the *interdependence problem.* To avoid this, we must use weak source input that does not contain enough information to precisely predict the target response.

For our work, we use contextual information (e.g. dialog history, belief state, database results) as a conditional source x. A common practice to embed contextual information is to use a GRU and a linear layer to encode, and as a result, we obtain continuous embeddings $\boldsymbol{x}$. However, continuous embeddings can result in the interdependence problem, since they can theoretically contain infinite information.

To weaken the encoder, we reduce the resolution of the input embeddings $\boldsymbol{x}$ by using discrete embeddings. We define it as an array of N-way categorical variables: $\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_M\}$, where each $\boldsymbol{x}_n$ is a N-sized binary vector and M is the number of variables. To obtain this, straight-through Gumbel-softmax (Jang et al., 2017) is applied to the conditional source encoded by a GRU and a linear layer. In our experiments in section 6, we will compare the results of these discrete embeddings with continuous embeddings.

4.2 Response Generation with Reinforcement Learning

To rearrange the invented hidden states of a neural HSMM decoder, we apply reinforcement learning, which we named this method as *phrase-level action reinforcement learning* (PHRASERL). Here, we consider a Markov decision process of input context as state $x \in \mathcal{S}$, hidden states (which represents phrases) as action space $z \in \mathcal{A}$, and task-success rate as rewards $r \in \mathcal{R}$. We define the timestep of hidden state selection as $t' = \{1, 2, \cdots, T'\}$. We consider the combined initial state selection and state transition as policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ and apply REINFORCE algorithm (Williams, 1992). For each reward $r_{t'}$ in time step t', we use discounted reward $G_{t'} = \sum_{k=0}^{T'} \gamma^k r_{t'+k}$ during training. Now, the policy gradient will be:

$$\nabla J(\pi) = \mathbb{E}\left[\sum_{t'=1}^{T'} \nabla_\pi \log \pi(z_{t'}|x) G_{t'}\right]. \quad (9)$$

Note that we do not train the GRU for the emission distribution; we only further train the hidden state transition. For embedding the contexts, we used the pre-trained encoder and did not further update during this RL step.

5 Experimental Settings

5.1 Task Description

For the experiments, we use MultiWOZ dataset (Budzianowski et al., 2018). MultiWOZ is a large-scale task-oriented dialog dataset, which contains seven types of domains such as booking restaurants, hotels, and train seats. We specifically use Dialogue-Context-to-Text Generation task proposed in the original paper. In this task, a model is given an oracle belief state, and the model's goal is to generate an appropriate and informative response. For the evaluation, we use BLEU, Inform Rate, and Success Rate. We also compute the total score, which is used in previous works to compare models in MultiWOZ dataset. Total score is calculated with $\mathrm{BLEU} + (\mathrm{Inform} + \mathrm{Success})/2$.

5.2 Model Details

Duplicated Hidden States While increasing the number of hidden states allows for a more expressive latent model, the computational complexity of the neural HSMM decoder will increase linearly depending on the number of hidden states K. In order to increase the number of hidden states without making the computation heavier, we use the same emission distribution for multiple hidden states as proposed in Wiseman et al. (2018). For instance, if we set the base state as 80 and duplicated 5 times, K will be 400 and we use $z \bmod 80$ for the input into the computation of emission distribution. This way, the model can utilize a large number of hidden states in the state transition, while the model only needs to run the GRU feed-forward for a smaller number of times to compute emission distribution.

Training Details We first train the neural HSMM decoder with supervised learning, and later we further train with reinforcement learning as explained in section 4.2. To embed the context information x, we use a MLP layer for encoding oracle belief state and a GRU for encoding dialog history. For comparison, we trained both continuous and discrete embeddings, which we denote each model as CONT and DISC respectively. To train with reinforcement learning, we use the MultiWOZ RL setup proposed in (Zhao et al., 2019). For the rewards, we use $r_{\text{success}} + r_{\text{inform}} + r_{\text{BLEU}}$.

The average loss is computed with the validation dataset after every epoch, and early stopping is performed after 5 consecutive epochs without improvement. When we determine the number of hidden states, we tested every 10 states from 40 to 120 for base states, and 1, 3, 5, 7 for duplication. In consequence, $K = 400$ (80 base states, duplicated 5 times) produced the best results, and the following evaluations are based on these results. For the vocabulary set, we substituted the words that occurred less than 30 times in the dataset with an unknown tag ($\langle unk \rangle$). We used beam search with a beam size of 5 for the decoding. For more details, refer to Appendix A.

5.3 Baseline and State-of-the-Art Models

Our model is compared with the following models:

- **Baseline** (Budzianowski et al., 2018) is proposed in the original MultiWOZ paper. The model is based on Seq2Seq with attention on the context words.

- **Word-Level RL** further trains the pre-trained baseline on reinforcement learning with the action space of words. It is known that this model often encounters a degeneration problem, in which the generated sentence will diverge from natural human sentences.

- **Latent Action Reinforcement Learning (LaRL)** (Zhao et al., 2019) introduces a discrete latent variable between the encoder and decoder to represent a dialog act. Similar to our model, it first trains on supervised pre-training, then it further trains the dialog policy with reinforcement learning.

- **Hierarchical Disentangled Self-Attention Network (HDSA)** (Chen et al., 2020) introduces hierarchical dialog act. The model is composed of 3 transformer layers which each layer corresponds to each hierarchy of the dialog act. The model switches the self-attention based on the dialog act, which is called the disentangled self-attention.

- **SOLOIST** (Peng et al., 2020) is a transformer-based auto-regressive language model for task-oriented dialog, pre-trained on large and diverse dialog corpora. The model is fine-tuned on MultiWOZ task.

- **MarCo** (Wang et al., 2020b) extends the idea of HDSA and considers a hierarchical dialog

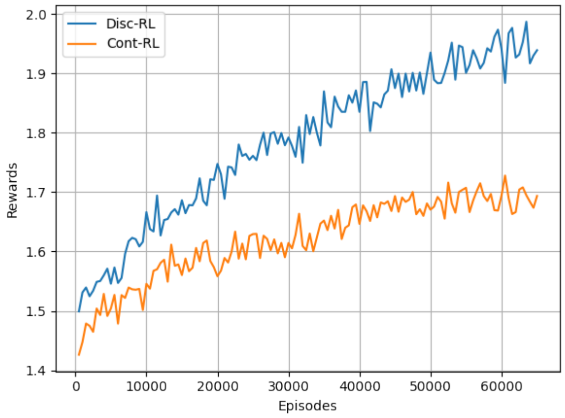

Figure 5: Average rewards for every 500 episodes of our proposed models. DISC-RL and CONT-RL indicates discrete and continuous embeddings, respectively. The maximum reward is 3.0.

act. The difference is that it co-generates the dialog act sequence and the response jointly.

- **HDNO** (Wang et al., 2020a) decouples the dialog policy and natural language generation by applying hierarchical RL. This model uses language model as a reward provider to maintain grammaticality.

Among these models, Word-Level RL, LaRL, and HDNO use reinforcement learning and, therefore, are considered as the main competitors to our PHRASERL. Also, note that HDNO uses rewards from external modules to avoid degeneration, but PhraseRL does not use them in this experiment.

6 Results and Analysis

6.1 Automatic Evaluations

Table 1 shows the results of the automatic evaluation of our models. We firstly see that applying reinforcement learning greatly improves the scores, indicating that hidden states had been an effective action space. We also see that discrete embeddings outperform continuous embeddings in every score. This shows that the model can improve generation performance by alleviating the interdependence problem with weaker encoders. It also can be observed that the discrete embeddings are significantly effective within reinforcement learning models. This can also be seen in the reward graph shown in figure 5, where discrete embeddings have a sharper reward increase compared to continuous embeddings. A possible reason behind this is that discrete embeddings led the phrase generation to be less diverse and strongly typed, which made the agent easier to learn the relation between a hidden state and the generated phrases.

We also compare our best model (DISC-RL) with the past works, and the results are shown in table 2. Our model achieved competitive results with the recently proposed state-of-the-art models, which also is near-human performance.

Comparing with LaRL, our model significantly outperforms in BLEU score even after applying RL. A possible reason for LaRL's low BLEU score is that it cannot fully express the diverse human sentences in a discrete global vector. On the other hand, PHRASERL can broaden the range of expression by dividing the action space into finer semantic units, which enables it to learn more human-like responses. Additionally, the improvement in Inform Rate and Success Rate can also be attributed to the ability of PHRASERL to flexibly select content.

Nevertheless, if we compare with state-of-the-art models without using RL, our model has a lower BLEU score. This suggests that using fixed phrases and arranging them have a drawback in regard to generating grammatical responses, since it lacks word-level flexibility. However, it is surprising to see that it still achieves a competitive score even with such disadvantage.

6.2 Model Analysis

Although our PHRASERL was able to maintain a high BLEU score, this can only be because the model was rewarded with the BLEU score during the training. We also trained the model without using the BLEU score for rewards and the results are shown in table 3. Although there is a slight decrease, it is still largely outperforming the Word-Level RL and LaRL. This indicates that our PHRASERL is resistant to degeneration to some extent, even without adding external modules for rewarding grammaticality as in HDNO. Further improvements can be expected by applying external rewards for avoiding degeneration, though this remains as future work.

Table 4 shows the generated phrases from randomly selected hidden states. By observing the outcome of DISC, we can notice that the common property of the generated phrases are interpretable: state 303 outputs ”verb phrases for the end of question”, state 239 outputs ”beginning of the question”, state 103 outputs ”features of a facility”, state 70 outputs ”back-channeling words”, and state 325

	BLEU (%)	Inform (%)	Success (%)	Total
CONT	15.3	56.7	42.9	65.2
DISC	15.7	71.5	46.3	74.6
CONT-RL	16.7	86.2	67.2	93.5
DISC-RL	**18.0**	**95.6**	**79.3**	**105.4**

Table 1: Evaluation results of MultiWOZ test dataset of our models. RL indicates additional training on reinforcement learning. The total score is computed with BLEU + (Inform + Success) / 2.

	RL	BLEU (%)	Inform (%)	Success (%)	Total
Human	-	-	91.0	82.7	-
Baseline (Budzianowski et al., 2018)	-	18.9	71.3	61.0	85.0
HDSA (Chen et al., 2020)	-	23.6	82.9	68.9	99.5
SOLOIST (Peng et al., 2020)	-	18.3	89.6	79.3	102.8
MarCo (Wang et al., 2020b)	-	20.0	92.3	78.6	105.5
Word-Level RL (Zhao et al., 2019)	✓	1.4	80.5	79.1	81.2
LaRL (Zhao et al., 2019)	✓	12.8	82.8	79.2	93.8
PHRASERL (Ours)	✓	18.0	95.6	79.3	105.4
HDNO (Wang et al., 2020a)	✓	18.9	96.4	84.7	109.5

Table 2: Evaluation results of MultiWOZ test dataset compared with previous works. Our PHRASERL is the result of DISC-RL. The results were obtained from corresponding papers.

outputs ”noun phrases for domains”. This indicates that the hidden states in Disc are strongly typed. Although phrases of CONT also seems to be typed, some states such as state 138 have multiple types. This is due to the interdependence problem because the RNN can recognize which type to use depending on the conditional input x.

6.3 Case Study

To verify that the hidden states are a valid and flexible action space, we qualitatively validate the generated phrases. Figure 6 shows the generated phrases from user input and possible responses that can be formed by reordering the phrases. The possible responses were generated from hidden states that were reordered by hand. We consider three criteria for a valid and flexible action space in MultiWOZ dataset: the model must be able to (1) inform appropriate content (e.g. restaurant name, departure time), (2) generate grammatical sentences, and (3) generate diverse dialog acts.

Content Appropriate contents for this case would be the area of the restaurant ([value_area]), price range ([value_pricerange]), name of the restaurant ([restaurant_name]), and type of food ([value_food]). These entities appear at least once in the generated phrases, and the model can select the content by acting on the hidden states. Furthermore, as shown in the third and fourth examples of possible responses, the model can use a similar hidden state sequence for generating similarly structured responses, but it can still tweak the content depending on their strategy.

We further investigated if the contents are sufficiently provided with the hidden states. We counted the number of cases in the test set where all the entities in the golden response were contained in the generated phrases. As a result, 83.6% of the cases had sufficient information in the generated phrases, which we consider enough because the model may have other response options. Therefore, we can conclude that the first condition has been met.

Grammar Although there remains the possibility of generating ungrammatical sentences, the possible responses in figure 6 show that an appropriate sequence of hidden states will allow generating fluent responses.

Dialog Act As shown in the bottom section in figure 6, the four response samples have different intentions. For instance, the first response asks the type of food to the user, while the second response recommend a restaurant and ask for a booking. Particularly, a second example contains two different dialog acts (recommend and offer booking), but the model can choose to finish with the first sentence to just recommend. This shows that our PHRASERL have disentangled representation for each dialog acts. Therefore, we can conclude that the model can flexibly select from several intentions.

Finally, in table 5, we show an example response

	BLEU (%)	Inform (%)	Success (%)	Total
$r_{\text{success}} + r_{\text{inform}} + r_{\text{BLEU}}$	18.0	95.6	79.3	105.4
$r_{\text{success}} + r_{\text{inform}}$	17.1	95.7	78.9	104.4

Table 3: Evaluation results of DISC-RL with different rewards in MultiWOZ test set.

CONT				
State 190	State 138	State 4	State 114	State 85
what type of attraction what day what area what type of food what information	to book a ticket a different type of to book it a hotel or to know	in the [area] located in the [area] at [time] free wifi free to enter	in the [area] . . that day . with that . available .	the address is [address] it leaves at [time] the postcode is [postcode] the price is [price] it is [pricerange]
DISC				
State 303	State 239	State 103	State 70	State 325
to stay ? ⟨eos⟩ to dine ? ⟨eos⟩ to dine in to book ? ⟨eos⟩ to arrive ? ⟨eos⟩	what day what time how many tickets what area what type of food	free internet and parking free to enter located at [address] [pricerange] -ly priced free wifi and parking	sure , okay , you are welcome . i am sorry , yes ,	any restaurant -s a lot of attractions any attractions any hotel -s a few

Table 4: Top-5 frequently generated phrases from randomly selected hidden states. ⟨eos⟩ indicates end-of-sentence.

User Input: I am looking for an [value_pricerange] restaurant in the center of town.

Generated Phrases:

Phrase	# of hidden states				
in the [value_area]	8	. would you like	2	a different area	1
?	6	[value_pricerange] restaurant -s	2	. it s	1
a good choice	5	there are no	2	. is there	1
.	4	there are [value_count]	2	a particular area	1
i have [value_count]	3	for you ?	1	. do you prefer	1
[value_food] , [value_food] ,	3	what type of food	1	i am sorry ,	1
price range	2	me to book [value_count]	1	. can i book	1
would you like	2	a few	1	and [restaurant_name]	1
to try	2	okay ,	1	preference for price range	1
a preference ?	2	do you have a	1	cuisine preference ?	1
i recommend [restaurant_name]	2	a table for you	1	. do you have	1
[value_food] or [value_food]	2	would you like me	1	. which	1
the [restaurant_name]	2	restaurant -s	1	would you be interested	1
to try a different	2	i have [value_count] options	1	there are [value_count] options	1
in the [value_pricerange]	2	you would like	1		

Possible Responses:

okay , i have [value_count] options . what type of food would you like ?
i recommend [restaurant_name] . can i book a table for you ?
there are no [value_pricerange] restaurant -s . would you like to try a different area ?
there are no [value_pricerange] restaurant -s . would you like to try a different price range ?

Figure 6: A case study for generated phrases of DISC. In the second section, we show the number of hidden states that output the same phrase. The third section shows possible responses which were generated from inputting sequence of hidden states ordered by hand. The colors indicate the corresponding phrases.

from MultiWOZ test dataset. We first see that CONT is not generating a grammatical response. This may be due to the interdependence problem that tries to output various sentences in the same hidden state sequence. On the other hand, the rest of our models can generate a grammatical and in-context response. We also observe that while models without RL generate plausible responses, RL models provide a more informative response by including hotel names and phone numbers. The reason is that the model often receives rewards by conveying those information.

7 Conclusion and Future Work

In conclusion, this paper proposes a phrase-level action reinforcement learning for neural response generation. A neural HSMM decoder is introduced to learn hidden states that output typed phrases, and we used them as the action space for reinforcement

Context	i am looking for a place to stay in the [value_area] of the city . i would prefer a [value_count] star hotel please .
Human Response	there are several guesthouses available . do you have a price preference ?
Baseline	i have [value_count] hotel -s in the [value_area] . would you like to book a room ?
Cont	yes , i have [value_count] a few in the [value_area] . would you like a hotel or guesthouse ?
Disc	sure , there are [value_count] options . is there a particular price range ?
Cont-RL	i have the [hotel_name] in the [value_area] . the phone number is [hotel_phone] .
Disc-RL	[hotel_name] is in the [value_area] . would you like me to book it for you ?

Table 5: Example responses from baseline and our models. ⟨unk⟩ represents unknown word of the model.

learning. Our experiments on MultiWOZ dataset have shown that the model is capable of generating flexible outputs with RL, achieved competitive results to state-of-the-art models. A possible future direction is to apply hierarchical reinforcement learning, where the high-level policy determines the dialog act of the entire response, while low-level policy designs the syntax of the response.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 21H03502. The authors like to thank Taichi Iki, Takuma Udagawa, and Taku Sakamoto for their constructive feedbacks.

References

Anusha Balakrishnan, Jinfeng Rao, Kartikeya Upasani, Michael White, and Rajen Subba. 2019. Constrained decoding for neural NLG from compositional representations in task-oriented dialogue. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 831–844, Florence, Italy. Association for Computational Linguistics.

Siqi Bao, Huang He, Fan Wang, Hua Wu, and Haifeng Wang. 2019. Plato: Pre-trained dialogue generation model with discrete latent variable. *arXiv preprint arXiv:1910.07931*.

Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. 2018. MultiWOZ - A large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *arXiv*, 1.

Wenhu Chen, Jianshu Chen, Pengda Qin, Xifeng Yan, and William Yang Wang. 2020. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 3696–3709.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pages 1–13.

Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. Deal or no deal? end-to-end learning of negotiation dialogues. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453.

Jiwei Li, Michel Galley, Chris Brockett, Georgios P. Spithourakis, Jianfeng Gao, and Bill Dolan. 2016a. A persona-based neural conversation model. *54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers*, 2:994–1003.

Jiwei Li, Will Monroe, Alan Ritter, Michel Galley, Jianfeng Gao, and Dan Jurafsky. 2016b. Deep reinforcement learning for dialogue generation. *arXiv preprint arXiv:1606.01541*.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Tsung-Hsien Wen, Blaise Thomson, and Steve Young. 2017. Neural belief tracker: Data-driven dialogue state tracking. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788.

Kevin P Murphy. 2002. Hidden semi-markov models (hsmms). *unpublished notes*, 2.

Baolin Peng, Chunyuan Li, Jinchao Li, Shahin Shayandeh, Lars Liden, and Jianfeng Gao. 2020. SOLOIST: Few-shot task-oriented dialog with a single pretrained auto-regressive model. *arXiv*.

Baolin Peng, Xiujun Li, Jianfeng Gao, Jingjing Liu, Kam-Fai Wong, and Shang-Yu Su. 2018. Deep dyna-q: Integrating planning for task-completion dialogue policy learning. *arXiv preprint arXiv:1801.06176*.

Osman Ramadan, Paweł Budzianowski, and Milica Gasic. 2018. Large-scale multi-domain belief tracking with knowledge sharing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 432–437.

Abhinav Rastogi, Xiaoxue Zang, Srinivas Sunkara, Raghav Gupta, and Pranav Khaitan. 2020. Towards scalable multi-domain conversational agents: The schema-guided dialogue dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8689–8696.

Abdelrhman Saleh, Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, and Rosalind Picard. 2019. Hierarchical Reinforcement Learning for Open-Domain Dialog. *arXiv*.

Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. A hierarchical latent variable encoder-decoder model for generating dialogues. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pages 3295–3301.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational linguistics*, 26(3):339–373.

Van-Khanh Tran and Le-Minh Nguyen. 2017. Natural language generation for spoken dialogue system using RNN encoder-decoder networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 442–451, Vancouver, Canada. Association for Computational Linguistics.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *arXiv preprint arXiv:1506.05869*.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. 2020a. Modelling hierarchical structure between dialogue policy and natural language generator with option framework for task-oriented dialogue system. *arXiv preprint arXiv:2006.06814*.

Kai Wang, Junfeng Tian, Rui Wang, Xiaojun Quan, and Jianxing Yu. 2020b. Multi-domain dialogue acts and response co-generation. *arXiv preprint arXiv:2004.12363*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Sam Wiseman, Stuart M. Shieber, and Alexander M. Rush. 2018. Learning Neural Templates for Text Generation. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pages 3174–3187.

Steve Young. 2006. Using pomdps for dialog management. *2006 IEEE ACL Spoken Language Technology Workshop, SLT 2006, Proceedings*, pages 8–13.

Tiancheng Zhao, Kaige Xie, and Maxine Eskenazi. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1208–1218.

Hao Zhou, Minlie Huang, and Xiaoyan Zhu. 2016. Context-aware natural language generation for spoken dialogue systems. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2032–2041, Osaka, Japan. The COLING 2016 Organizing Committee.

A Training Details

Table 6 shows the parameter used for the experiments.

Supervised Learning	
Batch Size	32
Word Embedding	128
State Embedding	128
Context Encoder RNN	GRU (256)
Decoder RNN	GRU (256)
Optimizer	Adam (lr=1e-3)
Dropout	0.5
Transition Matrix	$m_1 = 64, m_2 = 32$
Number of Hidden States K	400 (80 base states duplicated 5 times)
Maximum Emission Length L	4
Categorical Embeddings $M \times N$ (DISC Only)	10×10
Gumbel Softmax Temp. τ (DISC Only)	1.0
Reinforcement Learning	
Discount Rate γ	0.99
Optimizer	Adam(lr=1e-4)

Table 6: Parameters used in the experiments.