Generalizable and Explainable Dialogue Generation via Explicit Action Learning

Xinting Huang,[1] Jianzhong Qi,[1] Yu Sun,[2] Rui Zhang[1]*
[1]The University of Melbourne, [2]Twitter Inc.
{xintingh@student., jianzhong.qi@, rui.zhang@}unimelb.edu.au, ysun@twitter.com

Abstract

Response generation for task-oriented dialogues implicitly optimizes two objectives at the same time: task completion and language quality. Conditioned response generation serves as an effective approach to separately and better optimize these two objectives. Such an approach relies on system action annotations which are expensive to obtain. To alleviate the need of action annotations, latent action learning is introduced to map each utterance to a latent representation. However, this approach is prone to over-dependence on the training data, and the generalization capability is thus restricted. To address this issue, we propose to learn *natural language actions* that represent utterances as a span of words. This explicit action representation promotes generalization via the compositional structure of language. It also enables an explainable generation process. Our proposed unsupervised approach learns a memory component to summarize system utterances into a short span of words. To further promote a compact action representation, we propose an auxiliary task that restores state annotations as the summarized dialogue context using the memory component. Our proposed approach outperforms latent action baselines on MultiWOZ, a benchmark multi-domain dataset.

1 Introduction

Task-oriented dialogue systems complete tasks for users, such as making a hotel reservation or finding train routes, in a multi-turn conversation (Gao et al., 2018; Sun et al., 2016, 2017). The generated system utterances should not only be naturally sound, but more importantly be *informative*, i.e., to proceed the dialogue towards task completion. To fulfill this requirement, conditioned response generation is widely adopted based on system actions (Wen et al., 2017; Chen et al., 2019). The response generation process is decoupled into two consecutive steps, where an action is first selected and then an utterance is generated conditioned on this action. One can optimize each step towards its goal, i.e., informative and naturally sound, without impinging the other (Yarats and Lewis, 2018). However, such approaches rely on action annotations (as in Table 1), which require domain knowledge and extensive efforts to obtain.

Table 1: Conditioned Response Generation Example

Dialog Context	**Utterance**	
	User: *I need a train that departs bishops stortford on wednesday, please.*	
	Dialogue state annotation	
	`Train:{departure=bishops stortford, day=wednesday}`	
Action Selection	**System action annotation**	
	`train-inform:{choice=five}; train-request:{leaveat}`	
	Latent action	**Natural language action**
	[0,0,0,1,0]	{*'option', 'five' 'request','time'* }
Response Generation	*alright I found five options available. when would you like to leave by?*	

To deal with the absence of action annotations, latent action learning has been introduced (Zhao et al., 2018; Yarats and Lewis, 2018). System utterances are represented as low-dimensional latent variables by an auto-encoding task (Zhao et al., 2019), and utterances with the same representations are considered to convey similar meanings. Such action representations might be prone to over-dependence on the training data, which restricts the model generalization capability, especially when multiple domains are considered. This is because, without explicit supervision, the desired property of capturing the intentions of system utterances in the latent space cannot be enforced (Locatello

*Rui Zhang is the corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2020, pages 3981–3991
November 16 - 20, 2020. ©2020 Association for Computational Linguistics

et al., 2019), which in turn is due to the implicit nature of latent variables. For example, variational auto-encoder (VAE), which is often used for latent action learning, tends to produce a balanced distribution over the latent variables (Zhao et al., 2018), while the true distribution of system actions is highly imbalanced (Budzianowski et al., 2018). The resulting misaligned action representations would confuse the model of both steps and degenerate the sample efficiency in training.

To address the above issues, we propose to learn *natural language actions* that represent system utterances as a span of words, which explicitly reveal the underlying intentions. Natural language provides unique compositional structure while retaining the representation flexibility. These properties promote model generalization and thus make natural language a flexible representation for capturing characteristics with minimal assumptions (Jiang et al., 2019). Motivated by these advantages, we learn natural language actions by identifying *salient words* of system utterances. Salient refers to indicative for a prediction task (e.g., sentiment analysis) that takes as input the original utterance. The main rationale is that the principal information that the task concerns can be preserved by just the salient words. For example, the sentiment of sentence "*The movie starts out as competent but turn bland*" can be revealed by the word "*bland*" when it is identified salient by considering the complete context. In our scenarios, we consider measuring word saliency in terms of state transitions. This is because state transitions reflect how the intentions of a system utterance influence the dialogue progress, and action representations that capture such influences can well reveal the intentions (Chandak et al., 2019; Tennenholtz and Mannor, 2019; Huang et al., 2020b). By considering salient words for state tracking tasks as actions, we obtain action representations that enjoy the merits of natural language and indeed capture the characteristics of interest, i.e., intentions of system utterances.

Obtaining salient words by applying existing saliency identification approaches (Ribeiro et al., 2018) is, however, unable to produce *unified* action representations. Specifically, system utterances with the same intention might not share similar wordings, and existing attribution approaches can only identify salient words within utterances. We tackle this challenge by proposing a memory-augmented saliency approach that identifies salient words from a broader vocabulary. The vocabulary consists of all the words that could compose natural language actions, [1] and each word is stored as a slot in the memory component. By incorporating the memory component into a dialogue state tracking model, we use each system utterance as a query to perform memory retrieval, and the retrieval results are considered as salient words. The retrieval results might contain words that are redundant since we do not have direct supervision for the retrieval operations. For example, the resulting salient words might be "*but turn bland*" in the example shown earlier, which include unnecessary words and may lead to degenerated action results. To obtain *compact* action representations, we propose an auxiliary task based on pseudo parallel corpus, i.e., dialogue context and state annotation pairs. We observe that dialogue states serve as good examples of how compact representation should be. Therefore, we use the encoded dialogue context as query and ask the memory component to reconstruct its text-based dialogue states. In this way, the obtained concise actions generalize better and can be easily interpreted.

Our contributions are summarized as follows:

- We propose to learn explicit action representations (in contrast to latent action representations) for task-oriented dialogues, which promotes more generalizable and explainable dialogue generation.
- We propose a novel memory based approach with a pseudo parallel training scheme to obtain unified and compact action representations.
- We conduct experiments on a benchmark multi-domain dataset. Results show that our approach outperforms the state-of-the-art on both in-domain and cross-domain settings.

2 Preliminaries

Let $\{d_i|1 \leq i \leq N\}$ be a set of dialogues, and each dialogue contains n_d turns: $d_i = \{(c_t, a_t, x_t)|1 \leq t \leq n_d\}$, where c_t is the context at turn t, and a_t is the dialogue action of system utterance x_t. The context $c_t = \{u_1, x_1, ..., u_t\}$ consists of the dialogue history of user utterances u and system utterances x. Conditioned response generation tackles the context-to-response generation problem $p(x|c)$ via two consecutive steps: a content planning step decides a dialogue action to proceed the dialogues $p_l(a|c)$; and a surface realization step further trans-

[1] We consider content words from state annotations and task descriptions, which will be specified in Sec. 3.2

forms the decided action into naturally sound utterances $p_r(x|a,c)$. Using the two-step process, response generation could be optimized towards better task completion while maintaining high-quality language quality (Huang et al., 2020a; Zhao et al., 2019). The optimization process also consists of two parts. Firstly, context-action pairs are used to train the content planning model $p_l(a|c)$ using the cross-entropy loss.

$$\mathcal{L}_{act} = \sum_{d_i} \sum_{t=1:n_d} -\log(a_t^\top \cdot p_l(a|c_t)) \quad (1)$$

Then, the surface realization model $p_r(x|a,c)$ is optimized from the (c_t, a_t, x_t) triples to maximize the likelihood of ground-truth responses

$$\mathcal{L}_{lan} = \sum_{d_i} \sum_{t=1:n_d} -\log p_r(x_t|c_t, a_t) \quad (2)$$

Furthermore, to achieve better task completion, reinforcement learning (RL) is adopted to boost the pre-trained supervised models (Yarats and Lewis, 2018; Zhao et al., 2019). The rewards in terms of task completion (e.g., success rate) is usually computed based on the final generated response (Budzianowski et al., 2018). To avoid divergence from fluent utterances, this fine-tuning stage focuses on the content planning model $p_l(a|c)$ and keeps the parameters of $p_r(x|a,c)$ fixed. The reward R_t at each turn is back-propagated via policy gradients as:

$$\nabla_\phi \mathcal{J}(\phi) = \sum_{t=1:n_d} R_t \cdot \nabla_\phi \log p_l(a|c_t) \quad (3)$$

where ϕ denotes the parameters of model p_l.

In order to enable conditioned response generation when action annotations are absent, latent action learning is introduced. Given dialogues $\{(c_t, x_t)|1 \leq t \leq n_d\}$, latent action learning aims to map each utterance to a latent representation $z_d(x)$, e.g, one-hot (Wen et al., 2017), or continuous (Zhao et al., 2017). Based on the obtained $(c_t, z_d(x_t), x_t)$ triples, conditioned response generation is run as mentioned above. Existing latent action learning approaches mostly build on the idea of variational inference, where a latent space is found to reconstruct system utterances and thus encodes the main characteristics of utterances (Zhao et al., 2018; Huang et al., 2020a). The action representations learned from the latent space are, however, difficult to generalize due to the implicit nature and thus cause the sample inefficiency issue.

3 Proposed Model

3.1 Overview

We study the problem of *natural language action* learning for task-oriented dialogues. Specifically, we aim to represent each system utterance x_t as a sequence of word tokens $l(x_t) = [w_1, w_2, ..., w_n]$ without dialogue action annotations. The conditioned response generation is then performed using the enriched dialogues $\{(c_t, l(x_t), x_t)|1 \leq t \leq n_d\}$. Since natural language actions (i.e., sequences of tokens) encode the intention of system utterances in a compact and expressive way, both dialogue planning and language generation could achieve an improved generalization capability.

We design a memory component to identify the salient words of system utterances in terms of modeling state transitions (Sec. 3.2). To further boost the memory's capability in learning compact natural language actions, we propose a novel auxiliary task to identify salient words of dialogue context in a supervised setting (Sec. 3.3). Furthermore,we propose to take more advantage from the action learning phase by reusing the memory component for conditioned response generation (Sec. 3.4).

3.2 Memory Augmented Action Learning

We aim to obtain salient words that are indicative for the effects of system utterances in state transition. To model the such effects, we train a dialogue state tracking model that takes as input the system utterances. We then regard the sequence of words that substitute the system utterance and get similar state tracking results as salient words. To obtain sequences of words (i.e., natural language actions) of such characteristics, we use a learnable memory component that stores all potential words to form action representations, and optimize the memory in a self-supervised way.

Before presenting the proposed action learning approach, we first briefly introduce dialogue state tracking tasks. For dialogues $\{(c_t, x_t)|1 \leq t \leq n_d\}$, let $\{b_t|1 \leq t \leq n_d\}$ be the dialogue state for each turn, where $b_t \in \{0,1\}^{N_b}$ and N_b is the number of all slot-value pairs. Dialogue state tracking is usually formulated as a multi-label learning problem where the state at turn t predicted by modeling the conditional distribution $p(b_t|c_t) = p(b_t|u_t, x_{t-1}, b_{t-1})$, where b_{t-1} is the dialogue state in the previous turn. To model this conditional distribution, a state tracking model $p_\mathcal{B}(u_t, x_{t-1}, b_{t-1})$ mainly employs an utterance en-

coder, a context encoder to work with a slot-value predictor that estimates whether a slot-value pair should be included in the dialogue states (Lee et al., 2019). Specifically, the predictor takes as input a slot-value pair (s_i, e_i), and the encoded utterances $h^{utt} \in \mathbb{R}^D$ and context $h^{ctx} \in \mathbb{R}^D$ from the utterance encoder $f_{utt}(u_t, x_{t-1})$ and context encoder $f_{ctx}(b_{t-1})$ respectively, and D is the hidden dimension. The prediction is then performed by aggregating the results of slot-value predictor $f_{val}(h^{utt}, h^{ctx}, (s_i, e_i))$ for the complete N_b slot-value pairs. We optimize the state tracking model using the cross-entropy loss:

$$\mathcal{L} = \sum_{d_i} \sum_{t=1:n_d} -\log(b_t^\top \cdot p_{\mathcal{B}}(u_t, x_{t-1}, c_{t-1})) \quad (4)$$

where the parameters of $p_{\mathcal{B}}$, which include f_{utt}, f_{ctx}, and f_{val}, are jointly trained.

Based on the learned state tracking model, a straightforward idea of obtaining salient words is to apply importance attribution approaches. Specifically, these approaches measure the importance of each word by observing the prediction difference caused by replacing it (Ribeiro et al., 2018; Jin et al., 2020). As discussed before, this would result in different action representations for utterances with the same action. To address this issue, we consider learning action representations from a broader vocabulary, which releases the constraint of selecting salient words only within utterances.

Key-Value Memory Component

To this aim, we propose to use a memory component as the additional vocabulary. Note that the selection of words to build the vocabulary is task dependent, and we select the words appearing in state annotations and content words [2] extracted from task descriptions provided in the dataset (Budzianowski et al., 2018). This simple strategy is intuitive and turns out to be empirically competitive.

Given the built vocabulary, we adopt a key-value memory bank, where each memory slot stores a word included in the vocabulary. Each memory slot is associated with a key vector and a value vector, given by learnable matrix $K \in \mathbb{R}^{D \times N_v}$ and $V \in \mathbb{R}^{D \times N_v}$ respectively, where N_v is the number of words stored in the memory. The memory is utilized to obtain action representations by *context-aware memory retrieval*. Specifically, we regard the encoded utterance h_{utt} from the trained dialogue state tracking model as the query vector $q \in \mathbb{R}^D$. The retrieval is then conducted by computing the attention weights as

$$z = \text{softmax}(q^\top \cdot K) \quad (5)$$

where $z \in \mathbb{R}^{N_v}$ is a probability vector over the slots. Memory slots with higher probability indicate that the corresponding words are expected to be more salient to represent the system utterance. We could assume a natural language action $l(x_{t-1})$ containing k words is sampled k times from a categorical distribution given by z without replacement, where the value of k is set as a hyper-parameter.

Multi-Hop Mechanism

Building on the above sampling strategy, we further recognize that it is not plausible to assume natural language actions are of the same length by setting k as a hyper-parameter. This is because the conveyed information of system utterances can vary from each other. It is common to see certain utterances expressing more intentions than others, especially those directly determine task completion after information is accumulated. Thus, inspired by end-to-end memory network (Sukhbaatar et al., 2015), we design a multi-hop mechanism to adaptively decide the length of natural language actions. Specifically, after obtaining the probability vector z, we update the query based on the original query q and a weighted sum of memory values:

$$q^2 = q^1 + z^\top \cdot V \quad (6)$$

where $V \in \mathbb{R}^{D \times N_v}$ is the memory value matrix. Note that we denote the initial query vector q (i.e., h^{utt}) as q^1 for simplicity. Using query q^2, we could get a retrieval result z^2 as the same way in Eqn. 5.

By conducting such k-hop memory operation (i.e., k times retrieval using corresponding updated queries), we obtain k different categorical distributions. We now assume that each word is sampled from one distribution accordingly, and the length of natural language actions is indeed the number of hops carried out. Thus, by adaptively deciding the number of hops, we could learn variable length natural language actions. To this aim, we design an action gate component that predicts whether to carry out a next retrieval based on the current updated query. We perform such prediction based on the updated query, since it aggregates the information of former query and memory slots after every

[2] We consider nouns, verbs, and adjectives as content words.

retrieval operation. More specifically, we formulate the action gate as a binary random variable t, and its distribution is modeled as:

$$p_{gate}(t|z^1, z^2, .., z^{k-1}) = \sigma(G^\top \cdot q^k) \quad (7)$$

where $\sigma(\cdot)$ is the sigmoid function, and $G \in \mathbb{R}^D$ is a learnable vector. In this way, we can obtain natural language actions of appropriate length, which are sampled from the distributions obtained before the action gate indicate a stop of retrieving.

Training

The memory component and action gate are end-to-end trained in a self-supervised way, where the feedback is whether an utterance and its action representation lead to similar state transitions, We can measure such similarity using a dialogue state tracking (DST) model. However, a direct application of the DST model trained by Eqn. 4 might be prone to attribute changes between original utterances and compact natural language actions, which results in insufficient feedback. To address this issue, we adopt a denoising training strategy inspired by unsupervised machine translation (Lample et al., 2018, 2019), and obtain a DST model that is more robust to the attribute transformation. Specifically, we apply a noise function $g(x)$ to the utterances, and modify the DST model training loss as:

$$\mathcal{L}_{dst} = \sum_{d_i} -\log(b_t^\top \cdot p_{\mathcal{B}}(g(u_t), g(x_{t-1}), c_{t-1})) \quad (8)$$

where the noise function corrupts the input utterance by performing word drops and word order shuffling as specified in Lample et al. (2018).

With a slight abuse of notations, we use $p_{\mathcal{B}}(x_{t-1})$ to denote $p_{\mathcal{B}}(u_t, x_{t-1}, c_{t-1})$. We formulate the training loss for self-supervised task as:

$$\mathcal{L}_{mem} = \sum_{d_i} \Big(-\log(b_t^\top \cdot p_{\mathcal{B}}(\tilde{l}(x_{t-1}))) - \mathrm{KL}(p_{\mathcal{B}}(x_{t-1}) || p_{\mathcal{B}}(\tilde{l}(x_{t-1}))) \Big) \quad (9)$$

where KL is Kullback-Leibler divergence, and $\tilde{l}(x_{t-1}))$ is the natural language action obtained via the memory component. This loss enforces the learned action representations to restore both the ground truth and predicted state transitions. Note that the natural language actions are sampled from categorical distributions, which is non-differentiable. To get gradients for the memory component during back-propagation, we apply a continuous approximation, i.e., using gumbel-softmax trick instead to conduct sampling (Jang et al., 2016), to enable end-to-end differentiability.

3.3 Learning with Pseudo Parallel Corpus

Recall that we aim to learn natural language actions that are not only expressive but also *compact*, i.e., only including words that encode system intentions. Although the memory based approach could identify salient words from a broader vocabulary, the identified words might degenerate to the words making up most of the original utterances, which introduces redundant words into action representations. To avoid such suboptimal scenarios, we propose a supervised auxiliary task to further regularize the memory component. We use the encoded context h^{ctx} given by $f_{ctx}(b_t)$ from the dialogue state tracking model as query vectors, and attempt to recover the dialogue state from the memory component. Here, we consider word-based dialogue state representations instead of multi-hot representations, $b \in \{0, 1\}^{N_b}$. For example, the dialogue state "*food= european, price-range=moderate*" is transformed to a text span *['food', 'european', 'price-range','moderate']*. We then form a pseudo parallel corpus by pairing the word-based dialogue states and the corresponding encoded states as

$$\mathcal{P} = \{\langle f_{ctx}(b_t), b_t^{\text{text}}. \rangle | 1 \le t \le n_d\}$$

where b_t^{text} is the text span for b_t. We train the memory component using the pseudo parallel corpus as:

$$\mathcal{L}_{par} = \sum_{(h,b)\in\mathcal{P}} \Big(-\log p(b_t^{\text{text}} | z^{1,2,..k(b)}) + \sum_{2\le i\le k(b)} \text{cross-entropy}(g_i, p_{gate}(q^i)) \Big) \quad (10)$$

where $k(b)$ is the length of the text span b_t^{text}, and $g_i \in \{0, 1\}$ indicates whether the multi-hop operation should end at step i, and only take value one when i equals $k(b)$. For each pair in $\mathcal{P}$, the loss consists of two terms: the first one further guides the memory component to identify salient words; meanwhile the second term enforces the memory component to *only* pick salient words and promotes action representations to remain compact.

The overall training objective function of the natural language action learning is:

$$\mathcal{L} = \mathcal{L}_{mem} + \alpha\mathcal{L}_{par} + \beta\mathcal{L}_{dst} \quad (11)$$

where α and β are hyper-parameters. The reason we include the term $\mathcal{L}_{dst}$ during action learning is to ensure the DST model provides sufficient supervision. Some components in the DST model (i.e., f_{utt} and f_{ctx}) are updated via $\mathcal{L}_{mem}$ and $\mathcal{L}_{par}$, and by considering $\mathcal{L}_{dst}$, we could avoid a divergence from the state tracking task.

3.4 Conditional Response Generation

After obtaining natural language actions, we enrich the dialogues as $\{(c_t, l(x_t), x_t)|1 \leq t \leq n_d\}$, where $l(x_t)$ is the natural language action of utterance x_t. We could then run conditioned response generation to train content planning and language generation models as Eqn. 1-3. The learning efficiency can be improved by the more compact and noise-free action space. Moreover, the natural language actions present abundant information of correlations among actions, which allows for better generalization over actions (Chandak et al., 2019; Hu et al., 2019).

To further enhance the generalization capability and boost the learning efficiency, we consider re-use the memory component for conditioned response generation. Specifically, we focus on the content planning model $p_l(a|c)$, which aims to decide one natural language action from the action set for response generation [3]. We could implement the content planning model as a network that encodes the dialogue context c into a hidden state of the same dimension as query vectors in the memory component. By using the encoded results as query for memory retrieval, we obtain a distribution given by the retrieval results. We then select the action of highest probability determined by the obtained distribution as model output. This fine-tuning approach could not only reduce the model complexity for content planning, but also better harvest the knowledge gained in action learning phase.

4 Experiments

To show the effectiveness of the proposed approach, *memory-augmented saliency with parallel corpus* (MASP), we experiment on two dialogue generation settings (Sec. 4.1). We compare against state-of-the-art approaches in both settings (Sec. 4.2). We analyze the effectiveness of MASP components under different supervision ratios, and discuss how explainable generation is achieved (Sec. 4.3).

[3]We can also tackle content planning by a generative model, and details are introduced in Sec 4.3

4.1 Settings

We use MultiWOZ (Budzianowski et al., 2018), a multi-domain human-human conversational dataset in our experiments. It contains in total 8438 dialogues spanning over seven domains, and each dialogue has 13.7 turns on average. We use the separation of training, validation and testing data as original MultiWOZ dataset. We use the evaluation metrics as Budzianowski et al. (2018) to measure dialogue task completion, which are how often the system provides a correct entity (**Inform**) and answers all the requested information (**Success**). We use **BLEU** (Papineni et al., 2002) to measure the language quality of generated responses.

We use a three-layer transformer (Vaswani et al., 2017) with a hidden size of 128 and 4 heads as our base model for content planning and response generation, i.e., $p_l(a|c)$ and $p_r(a, c)$, respectively. We use grid search to find the best hyperparameters for the models based on validation performance, which we use a combination of Inform, Success and BLEU scores to measure. We choose the embedding dimensionality d among {50, 75, 100, 150, 200}, the hyperparameters α and β in [0.01, 1.0].

We consider two settings to thoroughly evaluate the conditioned response generation: *multi-domain joint training* and *cross-domain response generation*. In the first setting, we train MASP and other baselines using different sizes of the training dialogues (20%/50%/full), and for the tasks using 20% or 50% of data, the distribution of dialogues across domains are kept the same as the full training set. In the cross-domain setting, we adopt a leave-one-out approach to evaluate the generalization ability via a more challenging few-shot learning task. Specifically, we use one domain as low-resource target domain (with only 1% of dialogues are available for training) while the others as source domains.

We compare with the following baselines that do not consider conditioned generation: (1) **Seq-to-Seq** (Budzianowski et al., 2018) implemented based on transformer (Vaswani et al., 2017); (2) **TSCP** (Lei et al., 2018); and two baselines that adopt latent action learning for conditioned generation: (3) **LaRL** (Zhao et al., 2019); (4) **MALA** (Huang et al., 2020a). Note that for these two approaches, we experiment with both discrete and continuous latent action representations. We also compare the full model **MASP** with its two variants: (1) **Post-hoc Saliency** obtains action representations via the importance attribution technique

Table 2: Multi-Domain Joint Training Results

Model		20% Training Data			50% Training Data			Full Training Data		
		Inform	Success	BLEU	Inform	Success	BLEU	Inform	Success	BLEU
w/o Action	Seq-to-Seq	52.4	44.2	11.9	61.6	50.2	16.4	71.2	59.9	18.8
	TSCP	54.8	47.3	12.7	66.0	52.7	15.6	76.2	64.5	17.2
Continuous Latent Action	LaRL	51.1	44.0	12.7	63.4	50.9	14.3	70.8	60.5	14.5
	MALA	55.1	50.5	14.1	72.8	63.4	17.4	84.1	73.7	18.6
Discrete Latent Action	LaRL	60.5	51.9	10.8	69.2	60.1	13.3	81.5	69.2	14.8
	MALA	63.5	56.2	11.1	74.1	65.0	17.1	85.0	76.2	20.1
Proposed	Post-hoc	62.8	52.4	13.7	68.0	57.9	17.2	75.4	62.4	19.6
	Memory-based	64.7	55.4	13.6	76.1	70.6	19.1	84.9	75.2	20.8
	MASP	**70.2**	**61.8**	**14.9**	**78.7**	**71.5**	**19.4**	**88.3**	**77.1**	**21.7**

* Note that Post-hoc and Memory-based denotes the two variants Post-hoc Saliency and Memory-based Saliency.

as Jin et al. (2020); (2) **Memory-based Saliency** employs the same memory component as MASP but trained without the pseudo parallel corpus.

4.2 Overall Results

Table 2 shows that MASP outperforms baselines in the multi-domain joint training setting. MASP achieves better dialogue task completion (measured by Inform and Success) and language quality (measured by BLEU), especially in low resource scenarios. For example, MASP (70.2) outperforms MALA (63.5) by 10.5% under Inform when having 20% training data. Meanwhile, we also find that the memory component and pseudo parallel enhanced training are essential for getting effective action representations. For example, Post-hoc Saliency (57.9) is outperformed by a large margin compared to MALA (65.0) under Success when having 50% training data, while MASP (71.5) achieves a performance 10% gain over MALA. This validates that the unified and compact characteristics are required for natural language actions to boost conditioned generation. We further find that the contribution of the memory component and pseudo parallel corpus may vary in different ratios of training data. For example, the memory component brings 11.9% and 3.0% improvements compared to Post-hoc Saliency under Inform when the ratio is 50% and 20% respectively, while the pseudo parallel corpus brings 3.4% and 8.5% improvements compared to Memory-based Saliency. This is largely because the memory component could easily degenerate to utterance restoration when available training data is less, and thus the regularization provided by pseudo parallel corpus is more desired.

For cross-domain setting, Table 3 includes three representative domains (*hotel*, *attraction*, and *train*), and the observations on other domains are consistent. [4] The results show that MASP significantly outperforms the baselines in each configuration. For example, MASP (39.2) outperforms MALA (33.9) by 15.6% under Inform in *hotel* domain. By comparing results of Memory-based Saliency and MALA in *attraction* and *train*, we find that without pseudo parallel corpus, natural language actions could still be competitive occasionally. We will conduct a detailed analysis in the next section. We also find that continuous latent action approaches achieve comparable results as their discrete counterparts, while the results are opposite in the joint training setting. For example, MALA with continuous action (41.9) is slightly outperformed by its discrete counterparts (42.2) under Success using *attraction* as target. This is largely because the challenging cross-domain task could result in many mis-assigned action labels, and continuous action representations can still preserve certain knowledge of similarities among actions.

4.3 Discussions

We first study the effects of different components of MASP in the cross-domain setting. We compare MASP and its two variants with MALA (discrete action) under different dialogue ratios in target domains. The results are shown in Fig. 1(a) and Fig. 1. We can see that Memory-based Saliency is more comparable to MALA when using *train* as target domain, especially when the dialogue ratio is low. This is largely because there are many shared knowledge of system intentions and state transitions between *taxi* and *train* domains, and the memory component could benefit from such knowledge via the dialogue state tracking model. On the other hand, for target domains that do not have

[4] We omit the results of Seq-to-seq and TSCP in the table which are all worse than the latent action approaches

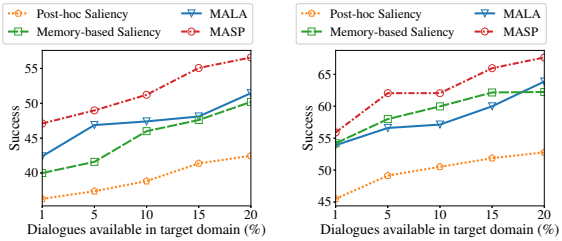
Table 3: Cross-Domain Generation Results

Model		Hotel			Attraction			Train		
		Inform	Success	BLEU	Inform	Success	BLEU	Inform	Success	BLEU
Continuous Latent Action	LaRL	26.7	22.3	11.4	35.3	30.4	13.1	40.3	36.4	13.1
	MALA	31.4	30.0	15.8	44.6	41.9	16.7	49.2	47.0	17.7
Discrete Latent Action	LaRL	24.1	22.7	9.1	35.8	30.0	11.8	43.2	40.9	12.8
	MALA	33.9	32.3	16.7	45.9	42.4	18.1	55.6	53.9	19.4
Proposed	Post-hoc	31.0	28.8	14.5	42.8	36.3	15.6	49.0	45.5	16.3
	Memory-based	34.7	32.0	14.6	43.3	40.0	17.2	57.6	54.2	18.3
	MASP	**39.2**	**35.1**	**17.2**	**52.5**	**47.1**	**18.6**	**59.2**	**55.9**	**19.4**

Table 4: Effects of Content Planning Model Design

Action	Planning Model	Inform	Success	BLEU
Post-hoc Saliency	Act-Dec	54.4	47.9	10.6
	Act-Cls	60.1	50.1	11.0
	Act-Cls (+emb)	62.8	52.4	13.7
MASP	Act-Dec	64.1	56.9	12.6
	Act-Dec (+mem)	68.0	58.6	14.2
	Act-Cls	64.3	57.7	11.4
	Act-Cls (+emb)	68.6	59.3	14.0
	Act-Cls (+mem)	**70.2**	**61.8**	**14.9**

* Results are in multi-domain joint training of 20% data.



(a) Attraction as target domain (b) Train as target domain

Figure 1: Effects of action characteristics

much advantage (e.g., *attraction*), the pseudo parallel corpus might contribute more to action learning. This conclusion is also consistent with what we observe in multi-domain joint training.

Last, we study the effects of content planning model design. We consider mainly two types of content planning model that works on natural language actions: action decoder and classifier, denoted as Act-Dec and Act-Cls, respectively. Specifically, an action decoder generates a text span and feed it to the language generation model, while action classifier conducts classification to select one action from the action set given by the training set. We also consider to enhance the planning model with (1) action embeddings computed by summing word-embedding of words in actions; (2) memory component as discussed in 3.4. From the results shown in Table 4, we can see that reusing the memory component could effectively improve the performance of conditioned response generation. We also find that action classifier generally perform better than action decoder, while the latter is more flexible to manipulate the content to generate. This is aligned with our intuition since more specific and task-relevant intentions are more favorable for task-oriented dialogues.

Moreover, through natural language actions, we could obtain a transparent response generation process, where the decided intermediate action is human-understandable. Such transparency could help alleviate the credit assigning issue by identifying the effectiveness of dialogue planning and surface realization. Table 5 shows that the proposed approach can obtain interpretable action representations (e.g., "request-departure") for the utterances that have the same intention but with different wording. This table also shows an error that our approach made in action learning, where the sentence highlighted in bold expresses "inform-address" instead of "inform-area". This might be caused by that the utterance contains multiple intentions and is thus more challenging for action learning. Table 6 shows that, with the learned natural language actions, we can better identify the source of errors in conditioned response generation. The two generated responses read naturally sound but express inappropriate intentions. The upper and lower examples showcase an action decision error and a language generation error, respectively. These help recognize the cause of errors and guide further optimization of the relevant components (content planning model or surface realization model).

5 Related Work

Early studies of conditioned response generation focus on enriching the meaning representations in task-oriented dialogues, e.g., utilizing graph structures and hierarchies among actions (Chen et al., 2019; Yang et al., 2020), decomposing into

Table 5: Natural Language Action Learning Results.

Natural language action	System utterances
{*'request', 'departure'*}	there are many trains during that time , where are you leaving from ?
	i am sorry , to help narrow down the results please reply with where you will be departing from
	i am going to need a little more information from you . where will you be leaving from ?
{*'inform', 'price', 'area', 'offer', 'reservation'*}	**the address is 169 high street chesterton** and the price range is fairly expensive . would you like to make reservations ?
	i have one that is called saigon city . it 's more expensive and located in the north . can i make a reservation for you ?
	we have 14 indian restaurants in the expensive category . do you have any more information to narrow down the search ?

* The parts highlighted in bold is the missing information of action representation.

Table 6: Better Credit Assigning via Natural Language Actions

User Utterance	i would like to leave from kings lynn sometime after 10:00 .
Ground-truth Response	we have 14 trains leaving after 10:00 , what time would you prefer to arrive by ?
Selected Action	{*'inform', 'leaveat',* ***'request', 'departure'***}
Generated Response	i have train leaving after 10:00, **where would you like to depart?**
User Utterance	is there a restaurant in the center serving italian then ?
Ground-truth Response	there are several italian restaurants downtown , would you like me to pick for you ?
Selected Action	{*'inform', 'food', 'area'* }
Generated Response	there are several italian restaurants , **do you have a preference to the area ?**

* The parts highlighted in bold is the error of either content planning or language generation.

fine-grained actions (Shu et al., 2019), or encoding syntax attributes (Balakrishnan et al., 2019). Since these approaches often assume expensive action annotations, recent years have seen a growing interest in learning latent actions in an unsupervised way (Zhao et al., 2019; Huang et al., 2020a). These approaches build on either adversarial learning (Hu et al., 2017; Wang et al., 2018; Yang et al., 2018) or variational inference (Kingma and Welling, 2014) and encode all system utterances via a self-reconstruction task or distant supervision (Yarats and Lewis, 2018). Due to their implicit nature, latent actions are difficult to generalize, and we aim to overcome this limitation by learning explicit action representations.

Our study is also related to attribution approaches, which aims to find features or regions of input that are important for tasks. Different types of techniques, including gradient-based (Selvaraju et al., 2017) and post-hoc (Ribeiro et al., 2018), are applied for reinforcement learning (Mott et al., 2019), computer vision (Adebayo et al., 2018), and text classification (Jin et al., 2020). While these works focus on interpreting model behaviors, we aim to find salient words beyond input and utilize them as action representations.

6 Conclusions

We propose explicit action learning to achieve generalizable and interpretable dialogue generation. Our proposed model MASP learns unified and compact action representations. We propose a memory component that summarizes system utterances into natural language actions, i.e., spans of words from a unified vocabulary. We further introduce an auxiliary task to encourage natural language actions to only preserve task-relevant information. Experimental results confirm that MASP achieves better performance compared with the state-of-the-art in different settings, especially when supervision is limited. We plan to consider structural action representation learning that could convey more information as future work.

Acknowledgement

We would like to thank Xiaojie Wang for the insightful discussions. This work is supported by Australian Research Council (ARC) Discovery Project DP180102050.

References

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim. 2018. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515.

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 *Annual Meeting of the Association for Computational Linguistics*, pages 831–844.

Pawel Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Iñigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gasic. 2018. Multiwoz - a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *Conference on Empirical Methods in Natural Language Processing*, pages 5016–5026.

Yash Chandak, Georgios Theocharous, James Kostas, Scott Jordan, and Philip Thomas. 2019. Learning action representations for reinforcement learning. In *International Conference on Machine Learning*, pages 941–950.

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

Jianfeng Gao, Michel Galley, and Lihong Li. 2018. Neural approaches to conversational ai. *arXiv preprint arXiv:1809.08267*.

Hengyuan Hu, Denis Yarats, Qucheng Gong, Yuandong Tian, and Mike Lewis. 2019. Hierarchical decision making by generating and following natural language instructions. In *Advances in Neural Information Processing Systems*, pages 10025–10034.

Zhiting Hu, Zichao Yang, Xiaodan Liang, R. Salakhutdinov, and E. Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596.

Xinting Huang, Jianzhong Qi, Yu Sun, and Rui Zhang. 2020a. Mala: Cross-domain dialogue generation with action learning. In *AAAI Conference on Artificial Intelligence*, pages 7977–7984.

Xinting Huang, Jianzhong Qi, Yu Sun, and Rui Zhang. 2020b. Semi-supervised dialogue policy learning via stochastic reward estimation. In *Annual Meeting of the Association for Computational Linguistics*, pages 660–670.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Yiding Jiang, Shixiang Shane Gu, Kevin P Murphy, and Chelsea Finn. 2019. Language as an abstraction for hierarchical deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 9414–9426.

Xisen Jin, Junyi Du, Zhongyu Wei, Xiangyang Xue, and Xiang Ren. 2020. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. *ArXiv*, abs/1911.06194.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Guillaume Lample, Sandeep Subramanian, Eric Michael Smith, Ludovic Denoyer, Marc'Aurelio Ranzato, and Y-Lan Boureau. 2019. Multiple-attribute text rewriting. In *International Conference on Learning Representations*.

Hwaran Lee, Jinsik Lee, and Tae-Yoon Kim. 2019. SUMBT: Slot-utterance matching for universal and scalable belief tracking. In *Annual Meeting of the Association for Computational Linguistics*, pages 5478–5483.

Wenqiang Lei, Xisen Jin, Min-Yen Kan, Zhaochun Ren, Xiangnan He, and Dawei Yin. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *Annual Meeting of the Association for Computational Linguistics*, page 1437–1447.

Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. 2019. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124.

Alex Mott, Daniel Zoran, Mike Chrzanowski, Daan Wierstra, and Danilo Jimenez Rezende. 2019. Towards interpretable reinforcement learning using attention augmented agents. In *Advances in Neural Information Processing Systems*, pages 12350–12359.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-precision model-agnostic explanations. In *AAAI Conference on Artificial Intelligence*, pages 1527–1535.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626.

Lei Shu, Hu Xu, Bing Liu, and Piero Molino. 2019. Modeling multi-action policy for task-oriented dialogues. In *Conference on Empirical Methods in Natural Language Processing*, pages 1304–1310.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *Advances in Neural Information Processing Systems*, pages 2440–2448.

Yu Sun, Nicholas Jing Yuan, Yingzi Wang, Xing Xie, Kieran McDonald, and Rui Zhang. 2016. Contextual intent tracking for personal assistants. In *SIGKDD*, pages 273–282.

Yu Sun, Nicholas Jing Yuan, Xing Xie, Kieran McDonald, and Rui Zhang. 2017. Collaborative intent prediction with real-time contextual data. *ACM Transactions on Information Systems (TOIS)*, 35(4):1–33.

Guy Tennenholtz and S. Mannor. 2019. The natural language of actions. In *International Conference on Machine Learning*, pages 6196–6205.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Xiaojie Wang, Rui Zhang, Yu Sun, and Jianzhong Qi. 2018. Kdgan: knowledge distillation with generative adversarial networks. In *Advances in Neural Information Processing Systems*, pages 775–786.

Tsung-Hsien Wen, Yishu Miao, Phil Blunsom, and Steve Young. 2017. Latent intention dialogue models. In *International Conference on Machine Learning*, pages 3732–3741.

Shiquan Yang, Rui Zhang, and Sarah Erfani. 2020. Graphdialog: Integrating graph knowledge into end-to-end task-oriented dialogue systems. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Zichao Yang, Zhiting Hu, Chris Dyer, Eric P Xing, and Taylor Berg-Kirkpatrick. 2018. Unsupervised text style transfer using language models as discriminators. In *Advances in Neural Information Processing Systems*, pages 7287–7298.

Denis Yarats and Mike Lewis. 2018. Hierarchical text generation and planning for strategic dialogue. In *International Conference on Machine Learning*, pages 5591–5599.

Tiancheng Zhao, Kyusong Lee, and Maxine Eskenazi. 2018. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *Annual Meeting of the Association for Computational Linguistics*, pages 1098–1107.

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 *North American Chapter of the Association for Computational Linguistics*, pages 1208–1218.

Tiancheng Zhao, Ran Zhao, and Maxine Eskenazi. 2017. Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. *Annual Meeting of the Association for Computational Linguistics*, pages 654–664.