MALA: Cross-Domain Dialogue Generation with 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 involves two basic components: dialogue planning and surface realization. These two components, however, have a discrepancy in their objectives, i.e., task completion and language quality. To deal with such discrepancy, conditioned response generation has been introduced where the generation process is factorized into action decision and language generation via explicit action representations. To obtain action representations, recent studies learn *latent actions* in an unsupervised manner based on the utterance lexical similarity. Such an action learning approach is prone to diversities of language surfaces, which may impinge task completion and language quality. To address this issue, we propose *multi-stage adaptive latent action* learning (MALA) that learns *semantic latent actions* by distinguishing the effects of utterances on dialogue progress. We model the utterance effect using the transition of dialogue states caused by the utterance and develop a semantic similarity measurement that estimates whether utterances have similar effects. For learning semantic actions on domains without dialogue states, MALA extends the semantic similarity measurement across domains progressively, i.e., from aligning shared actions to learning domain-specific actions. Experiments using multi-domain datasets, SMD and MultiWOZ, show that our proposed model achieves consistent improvements over the baselines models in terms of both task completion and language quality.

1 Introduction

Task-oriented dialogue systems complete tasks for users, such as making a restaurant reservation or scheduling a meeting, in a multi-turn conversation (Gao, Galley, and Li 2018; Sun et al. 2016; 2017). Recently, end-to-end approaches based on neural encoder-decoder structure have shown promising results (Wen et al. 2017b; Madotto, Wu, and Fung 2018). However, such approaches directly map plain text dialogue context to responses (i.e., utterances), and do not distinguish two basic components for response generation: *dialogue planning* and *surface realization*. Here, dialogue planning means choosing an action (e.g., to request information such as the preferred cuisine from the user, or provide a restaurant recommendation to the user), and surface realization means transforming the chosen action into natural language responses. Studies show that not distinguishing these two components can be problematic since they have a discrepancy in objectives, and optimizing decision making on choosing actions might adversely affect the generated language quality (Yarats and Lewis 2018; Zhao, Xie, and Eskenazi 2019).

Table 1: System Utterance Action Example

System utterances	
Domain: Hotel (a). *Was there a particular section of town you were looking for?* (b). *Which area could you like the hotel to be located at?*	Domain: Attaction (c). *Did you have a particular type of attraction you were looking for?* (d). *great , what are you interested in doing or seeing ?*
System intention (ground truth action)	
`Request(Area)`	`Request(Type)`
Latent action (auto-encoding approach)	
(a): *[0,0,0,1,0]*; (b): *[0,1,0,0,0]*	(c): *[0,0,0,1,0]*; (d): *[0,0,0,0,1]*
Semantic latent action (proposed)	
(a) & (b): *[0,0,0,1,0]*	(c) & (d): *[0,0,0,0,1]*

To address this problem, conditioned response generation that relies on action representations has been introduced (Wen et al. 2015; Chen et al. 2019). Specifically, each system utterance is coupled with an explicit action representation, and responses with the same action representation convey similar meaning and represent the same action. In this way, the response generation is decoupled into two consecutive steps, and each component for conditioned response generation (i.e., dialogue planning or surface realization) can optimize for different objectives without impinging the other. Obtaining action representations is critical to conditioned response generation. Recent studies adopt variational autoencoder (VAE) to obtain low-dimensional latent variables that represent system utterances in an unsupervised way. Such an auto-encoding approach cannot effectively handle various types of surface realizations, especially when these exist multiple domains (e.g., hotel and attraction). This is because

*Rui Zhang is the corresponding author.

the latent variables learned in this way mainly rely on the lexical similarity among utterances instead of capturing the underlying intentions of those utterances. In Table 1, for example, system utterances (a) and (c) convey different intentions (i.e., `request(area)` and `request(type)`), but may have the same auto-encoding based latent action representation since they share similar wording.

To address the above issues, we propose a multi-stage approach to learn *semantic latent actions* that encode the underlying intention of system utterances instead of surface realization. The main idea is that the system utterances with the same underlying intention (e.g., `request(area)`) will lead to similar *dialogue state transitions*. This is because dialogue states summarize the dialogue progress towards task completion, and a dialogue state transition reflect how the intention of system utterance influences the progress at this turn. To encode underlying intention into semantic latent actions, we formulate a loss based on whether the reconstructed utterances from VAE cause similar state transitions as the input utterances. To distinguish the underlying intention among utterances more effectively, we further develop a regularization based on the similarity of resulting state transitions between two system utterances.

Learning the semantic latent actions requires annotations of the dialogue states. In many domains, there are simply no such annotations because they require extensive human efforts and are expensive to obtain. We tackle this challenge by transferring the knowledge of learned semantic latent actions from state annotation rich domains (i.e., source domains) to those without state annotation (i.e., target domains). We achieve knowledge transferring in a progressive way, and start with actions that exist on both the source and target domain, e.g., `Request(Price)` in both hotel and attraction domain. We call such actions as *shared actions* and actions only exist in the target domain as *domain-specific actions*. We observe that system utterances with shared actions will lead to similar states transitions despite belonging to different domains. Following this observation, we find and align the shared actions across domains. With action-utterance pairs gathered from the above shared actions aligning, we train a network to predict the similarity of resulting dialogue state transitions by taking as input only texts of system utterances. We then use such similarity prediction as supervision to better learn semantic latent actions for all utterances with domain-specific actions.

Our contributions are summarized as follows:

- We are the first to address the problem of cross-domain conditioned response generation without requiring action annotation.
- We propose a novel latent action learning approach for conditioned response generation which captures underlying intentions of system utterances beyond surface realization.
- We propose a novel multi-stage technique to extend the latent action learning to cross-domain scenarios via shared-action aligning and domain-specific action learning.
- We conduct extensive experiments on two multi-domain human-to-human conversational datasets. The results show the proposed model outperforms the state-of-the-art on both in-domain and cross-domain response generation settings.

2 Related Work

2.1 Controlled Text Generation

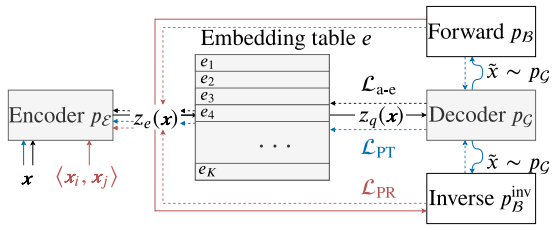
Controlled text generation aims to generate responses with controllable attributes. Many studies focus on open-domain dialogues' controllable attributes, e.g., style (Yang et al. 2018), sentiment (Shen et al. 2017), and specificity (Zhang et al. 2018). Different from open-domain, the controllable attributes for task-oriented dialogues are usually *system actions*, since it is important that system utterances convey clear intentions. Based on handcrafted system actions obtained from domain ontology, action-utterance pairs are used to learn semantically conditioned language generation models (Wen et al. 2015; Chen et al. 2019). Since it requires extensive efforts to build action sets and collect action labels for system utterances, recent years have seen a growing interest in learning utterance representations in an unsupervised way, i.e., *latent action learning* (Zhao, Lee, and Eskenazi 2018; Zhao, Xie, and Eskenazi 2019). Latent action learning adopts a pretraining phase to represent each utterance as a latent variable using a reconstruction based variational auto-encoder (Yarats and Lewis 2018). The obtained latent variable, however, mostly reflects lexical similarity and lacks sufficient semantics about the intention of system utterances. We utilize the dialogue state information to enhance the semantics of the learned latent actions.

2.2 Domain Adaptation for Task-oriented Dialogues

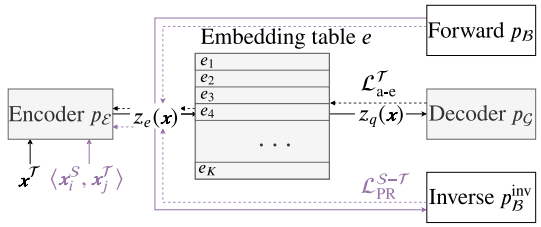
Domain adaptation aims to adapt a trained model to a new domain with a small amount of new data. This is studied in computer vision (Saito, Ushiku, and Harada 2017), item ranking (Wang et al. 2018a; Huang et al. 2019), and multi-label classification (Wang et al. 2018b; 2019; Sun and Wang 2019). For task-oriented dialogues, early studies focus on domain adaptation for individual components, e.g., intention determination (Chen, Hakkani-Tür, and He 2016), dialogue state tracking (Mrkšić et al. 2015), and dialogue policy (Mo et al. 2018; Yin et al. 2018). Two recent studies investigate end-to-end domain adaptation. DAML (Qian and Yu 2019) adopts model-agnostic meta-learning to learn a seq-to-seq dialogue model on target domains. ZSDG (Zhao and Eskenazi 2018) conducts adaptation based on action matching, and uses partial target domain system utterances as domain descriptions. These end-to-end domain adaption methods are either difficult to be adopted for conditioned generation or needing a full annotation of system actions. We aim to address these limitations in this study.

3 Preliminaries

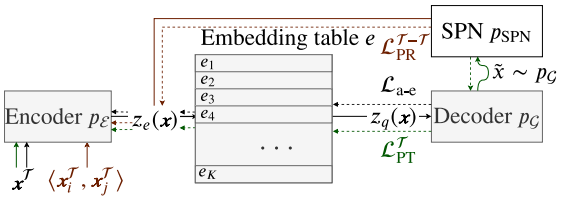
Let $\{d_i|1 \leq i \leq N\}$ be a set of dialogue data, and each dialogue d_i contains n_d turns: $d_i = \{(c_t, x_t)|1 \leq t \leq n_d\}$, where c_t and x_t are the context and system utterance at turn t, respectively. The context $c_t = \{u_1, x_1, ...u_t\}$ consists of the dialogue history of user utterances u and system utterances x. Latent action learning aims to map each system utterance x to a representation $z_d(x)$, where utterances with the same representation express the same action. The form of the representations $z_d(x)$ can be, e.g., one-hot (Wen et



(a) Stage-I: Semantic Latent Action Learning



(b) Stage-II: Action Alignment across Domains



(c) Stage-III: Domain-Specific Action Learning

Figure 1: Overall Framework of MALA.

al. 2015), multi-way categorical, and continuous (Zhao, Xie, and Eskenazi 2019). We use the one-hot representation due to its simplicity although the proposed approach can easily extend to other representation forms.

We obtain the one-hot representation via VQ-VAE, a discrete latent VAE model (van den Oord, Vinyals, and Koray 2017). Specifically, an encoder $p_{\mathcal{E}}$ encodes utterances as $z_e(x) \in \mathbb{R}^D$, and a decoder $p_{\mathcal{G}}$ reconstructs the original utterance based on inputs $z_q(x) \in \mathbb{R}^D$, where D is the hidden dimension. The difference lies in that between $z_e(x)$ and $z_q(x)$, we build a discretization bottleneck using a nearest-neighbor lookup on an embedding table $e \in \mathbb{R}^{K \times D}$ and obtain $z_q(x)$ by finding the embedding vector in e having the closest Euclidean distance to $z_e(x)$ i.e.,

$$z_q(x) = e_k \text{ where } k = \underset{j \in |K|}{\operatorname{argmin}} \|z_e(x) - e_j\|_2 .$$

The learned latent $z_d(x)$ is a one-hot vector that only has 1 at index k. All components, including $p_{\mathcal{E}}$, $p_{\mathcal{G}}$ and embedding table e, are jointly trained using auto-encoding objective as

$$\mathcal{L}_{\text{a-e}} = \mathbb{E}_x[-\log p_{\mathcal{G}}(x|z_q(x)) + \|z_e(x) - z_q(x)\|_2^2] \quad (1)$$

The structure of VQ-VAE is illustrated in Fig. 1(a), where the three components are marked in grey color.

4 Proposed Model

4.1 Overview

To achieve better conditioned response generation for task-oriented dialogues, we propose *multi-stage adaptive latent action learning* (MALA). Our proposed model works for two scenarios: (i) For domains with dialogue state annotations, we utilize these annotations to learn semantic latent actions to enhance the conditioned response generation. (ii) For domains without state annotations, we transfer the knowledge of semantic latent actions learned from the domains with rich annotations, and thus can also enhance the conditioned response generation for these domains.

The overall framework of MALA is illustrated in Fig. 1. The proposed model is built on VQ-VAE that contains encoder $p_{\mathcal{E}}$, embedding table e, and decoder $p_{\mathcal{G}}$. Besides auto-encoding based objective $\mathcal{L}_{\text{a-e}}$, we design pointwise loss $\mathcal{L}_{\text{PT}}$ and pairwise loss $\mathcal{L}_{\text{PR}}$ to enforce the latent actions to reflect underlying intentions of system utterances. For domains with state annotations (see Fig. 1a), we train $p_{\mathcal{B}}$ and $p_{\mathcal{B}}^{\text{inv}}$ to measure state transitions and develop the pointwise and pairwise loss (Sec. 4.2). For domains without state annotations (see Fig. 1b), we develop a pairwise loss $\mathcal{L}_{\text{PR}}^{\mathcal{S}-\mathcal{T}}$ based on $p_{\mathcal{B}}$ and $p_{\mathcal{B}}^{\text{inv}}$ from annotation-rich-domains. This loss measure state transitions for a cross-domain utterance pair, and thus can find and align shared actions across domains (Sec. 4.3). We then train a similarity prediction network p_{SPN} to substitute the role of state tracking models, which only taking as input raw text of utterances. We using p_{SPN} predictions as supervision to form pointwise $\mathcal{L}_{\text{PT}}^{\mathcal{T}-\mathcal{T}}$ and pairwise loss $\mathcal{L}_{\text{PR}}^{\mathcal{T}-\mathcal{T}}$ (see Fig. 1c), and thus obtain semantic latent actions for domain without state annotations (Sec. 4.4).

4.2 Stage-I: Semantic Latent Action Learning

We aim to learn semantic latent actions that align with the underlying intentions for system utterances. To effectively capture the underlying intention, we utilize dialogue state annotations and regard utterances that lead to similar state transition as having the same intention. We train dialogue state tracking model to measure whether any two utterance will lead to a similar state transition. We apply such measurement in (i) a pointwise manner, i.e., between a system utterance and its reconstructed counterpart from VAE, and (ii) a pairwise manner, i.e., between two system utterances.

Dialogue State Tracking Before presenting the proposed pointwise measure, we first briefly introduce dialogue state tracking tasks. Dialogue states (also known as dialogue belief) are in the form of predefined slot-value pairs. Dialogues with state (i.e., belief) annotations are represented as $d_i = \{(c_t, b_t, x_t)|1 \le t \le n_d\}$, where $b_t \in \{0,1\}^{N_b}$ is the dialogue state at turn t, and N_b is the number of all slot-value pairs. Dialogue state tracking (DST) is a multi-label learning process that models the conditional distribution $p(b_t|c_t) = p(b_t|u_t, x_{t-1}, c_{t-1})$. Using dialogue states annotations, we first train a state tracking model $p_{\mathcal{B}}$ with the following cross-entropy loss:

$$\mathcal{L} = \sum_{d_i} \sum_{t=1:n_d} -\log(b_t^\top \cdot p_{\mathcal{B}}(u_t, x_{t-1}, c_{t-1})) \quad (2)$$

$$p_{\mathcal{B}}(u_t, x_{t-1}, c_{t-1}) = \text{softmax}(h(u_t, x_{t-1}, c_{t-1}))$$

where $h(\cdot)$ is a scoring function and can be implemented in various ways, e.g., self attention models (Zhong, Xiong, and Socher 2018), or an encoder-decoder (Wu et al. 2019).

Pointwise Measure With the trained state tracking model $p_{\mathcal{B}}$, we now measure whether the reconstructed utterance output can lead to a similar dialogue state transition from turn $t-1$ to t (i.e., forward order). We formulate such measure as a cross-entropy loss between original state b_t and model $p_{\mathcal{B}}$ outputs when replacing system utterance x_{t-1} in inputs with $\tilde{x}_{t-1}$.

$$\mathcal{L}_{\text{fwd}} = \mathbb{E}_x[-\log(b_t^\top \cdot p_{\mathcal{B}}(b_t|u_t, \tilde{x}_{t-1}, c_{t-1}))] \quad (3)$$
$$\tilde{x}_{t-1} \sim p_{\mathcal{G}}(z_q(x_{t-1}))$$

where $\tilde{x}_{t-1}$ is sampled from the decoder output. Note that once state tracking model $p_{\mathcal{B}}$ finish training, its parameters will not be updated and $\mathcal{L}_{\text{fwd}}$ is only used for training the components of VAE, i.e., the encoder, decoder and the embedding table. To get gradients for these components during back-propagation, we apply a continuous approximation trick (Yang et al. 2018). Specifically, instead of feeding sampled utterances as input to state tracking models, we use Gumbel-softmax (Jang, Gu, and Poole 2016) distribution to sample instead. In this way outputs of the decoder $p_{\mathcal{G}}$ becomes a sequence of probability vectors, and we can use standard back-propagation to train the generator:

We expect the dialogue state transition in forward order can reflect the underlying intentions of system utterances. However, the state tracking model $p_{\mathcal{B}}$ heavily depends on user utterance u_t, meaning that shifts of system utterances intentions may not sufficiently influence the model outputs. This prevents the considered state transitions modeled from providing valid supervision for semantic latent action learning. To address this issue, inspired by inverse models in reinforcement learning (Pathak et al. 2017), we formulate a inverse state tracking to model the dialogue state transition from turn t to $t-1$. Since dialogue state at turn t already encodes information of user utterance u_t, we formulate the inverse state tracking as $p(b_{t-1}|x_{t-1}, b_t)$. In this way the system utterance plays a more important role in determining state transition. Specifically, we use state annotations to train an inverse state tracking model $p_{\mathcal{B}}^{\text{inv}}$ using the following cross-entropy loss:

$$\mathcal{L} = \sum_{d_i} \sum_{t=2:n_d} -\log(b_{t-1}^\top \cdot p_{\mathcal{B}}^{\text{inv}}(|x_{t-1}, b_t)) \quad (4)$$
$$p_{\mathcal{B}}^{\text{inv}}(x_{t-1}, b_t) = \text{softmax}(g(x_{t-1}, b_{t-1}))$$

where the scoring function $g(\cdot)$ can be implemented in the same structure as $h(\cdot)$. The parameters of inverse state tracking model $p_{\mathcal{B}}^{\text{inv}}$ also remain fixed once training is finished.

We use the inverse state tracking model to measure the similarity of dialogue state transitions caused by system utterance and its reconstructed counterpart. The formulation is similar to forward order:

$$\mathcal{L}_{\text{inv}} = \mathbb{E}_x[-\log(b_{t-1}^\top \cdot p_{\mathcal{B}}^{\text{inv}}(b_{t-1}|\tilde{x}_{t-1}, b_t))] \quad (5)$$
$$\tilde{x}_{t-1} \sim p_{\mathcal{G}}(z_q(x_{t-1}))$$

Thus, combining the dialogue state transitions modeled in both forward and inverse order, we get the full pointwise loss for learning semantic latent actions:

$$\mathcal{L}_{\text{PT}} = \mathcal{L}_{\text{fwd}} + \mathcal{L}_{\text{inv}} \quad (6)$$

Pairwise Measure To learn semantic latent actions that can distinguish utterances with different intentions, we further develop a pairwise measure that estimates whether two utterances lead to similar dialogue state transitions.

With a slight abuse of notation, we use x_i and x_j to denote two system utterances. We use u_i, c_i, b_i to denote the input user utterance, dialogue context, and dialogue state for dialogue state tracking models $p_{\mathcal{B}}$ and $p_{\mathcal{B}}^{\text{inv}}$, respectively. We formulate a pairwise measurement of state transitions as

$$s_{i,j} = s_{\text{fwd}}(x_i, x_j) + s_{\text{inv}}(x_i, x_j) \quad (7)$$
$$s_{\text{fwd}}(x_i, x_j) = \text{KL}(p_{\mathcal{B}}^{\text{fwd}}(u_i, x_i, c_i) \;||\; p_{\mathcal{B}}^{\text{fwd}}(u_i, x_j, c_i))$$
$$s_{\text{inv}}(x_i, x_j) = \text{KL}(p_{\mathcal{B}}^{\text{inv}}(x_i, b_i) \;||\; p_{\mathcal{B}}^{\text{inv}}(x_j, b_i))$$

where KL is the Kullback-Leibler divergence. Both $p_{\mathcal{B}}$ and $p_{\mathcal{B}}^{\text{inv}}$ take inputs related to x_i. We can understand $s_{i,j}$ in this way that it measures how similar the state tracking results are when replacing x_i with x_j as input to $p_{\mathcal{B}}$ and $p_{\mathcal{B}}^{\text{inv}}$.

To encode the pairwise measure into semantic latent action learning, we first organize all system utterances in a pairwise way $\mathcal{P} = \{\langle (x_i, x_j), s_{i,j} \rangle | 1 \le i, j \le N_u^{\mathcal{S}}\}$ where $N_u^{\mathcal{S}}$ is the total number of system utterances in the domains with state annotations. We then develop a pairwise loss to incorporate such measure on top of the VAE learning.

$$\mathcal{L}_{\text{PR}} = \sum_{\mathcal{P}} -s_{ij}^{\text{avg}} \log d(x_i, x_j) - (1 - s_{ij}^{\text{avg}}) \log(1 - d(x_i, x_j)) \quad (8)$$
$$d(x_i, x_j) = \sigma(-z_e(x_i)^\top z_e(x_j))$$

where σ is the sigmoid function, s_{ij}^{avg} is the average of $s_{i,j}$ and $s_{j,i}$, and $z_e(x) \in \mathbb{R}^{\mathbb{D}}$ is encoder $p_{\mathcal{E}}$ outputs. The pairwise loss $\mathcal{L}_{\text{PR}}$ trains $p_{\mathcal{E}}$ by enforcing its outputs of two system utterances to have far distances when these two utterance lead to different state transitions, and vice versa.

The overall objective function of the semantic action learning stage is:

$$\mathcal{L}_{\text{S-I}} = \mathcal{L}_{\text{a-e}} + \alpha \mathcal{L}_{\text{PT}} + \beta \mathcal{L}_{\text{PR}} \quad (9)$$

where α and β are hyper-parameters. We adopt $\mathcal{L}_{\text{S-I}}$ to train VAE with discretization bottleneck and obtain utterance-action pair (e.g., utterance (c) and its semantic latent action in Table 1) that encodes the underlying intentions for each system utterance in the domains with state annotations.

4.3 Stage-II: Action Alignment across Domains

In order to obtain utterance-action pairs in domains having no state annotations, we propose to progressively transfer the knowledge of semantic latent actions from those domains with rich state annotations. At this stage, we first learn semantic latent actions for the utterances that have co-existing intentions (i.e., shared actions) across domains.

We use $x^{\mathcal{S}}$ and $x^{\mathcal{T}}$ to denote system utterances in the source and target domain, respectively. The set of all utterances is denoted by:

$$U^{\mathcal{S}} = \{x_i^{\mathcal{S}} | 1 \le i \le N_u^{\mathcal{S}}\}; U^{\mathcal{T}} = \{x_j^{\mathcal{T}} | 1 \le j \le N_u^{\mathcal{T}}\}$$

where $N_u^{\mathcal{S}}$ and $N_u^{\mathcal{T}}$ are the total utterance number in each domain, respectively. We adopt the proposed pairwise measure to find the target domain system utterances that have

shared actions with the source domain. Based on the assumption that although from different domains, utterances with the same underlying intention are expected to lead to similar state transitions, we formulate the pairwise measure of cross-domain utterance pairs as:

$$s^c_{i,j} = s_{\text{fwd}}(x^{\mathcal{S}}_i, x^{\mathcal{T}}_j) + s_{\text{inv}}(x^{\mathcal{S}}_i, x^{\mathcal{T}}_j) \tag{10}$$

where s_{fwd} and s_{inv} are computed using the trained $p_{\mathcal{B}}$ and p^{inv}_B. Since it only requires the trained dialogue state tracking models and state annotations related to $x^{\mathcal{S}}_i$, this pairwise measure is asymmetrical. Taking advantage of the asymmetry, this cross-domain pairwise measure can still work when we only have raw texts of dialogues in the target domain.

We then utilize the cross-domain pairwise for action alignment during latent action learning on the target domain. We then formulate a loss Incorporating action alignment:

$$\begin{aligned}\mathcal{L}^{\mathcal{S}-\mathcal{T}}_{\text{PR}} = \sum_{x^S, x^T} & -s^c_{i,j} \log d(x^{\mathcal{S}}_i, x^{\mathcal{T}}_j) \\ & - (1 - s^c_{i,j}) \log(1 - d(x^{\mathcal{S}}_i, x^{\mathcal{T}}_j)) \\ d(x^{\mathcal{S}}_i, x^{\mathcal{T}}_j) &= \sigma(-z_e(x^{\mathcal{S}}_i)^\top z_e(x^{\mathcal{T}}_j))\end{aligned} \tag{11}$$

where $d(x^{\mathcal{S}}_i, x^{\mathcal{T}}_j)$ is computed based on outputs of the same encoder $p_{\mathcal{E}}$ from VAE at stage-I. We also use utterances in the target domain to formulate an auto-encoding loss:

$$\mathcal{L}^{\mathcal{T}}_{\text{a-e}} = \mathbb{E}_{x \in U^{\mathcal{T}}}[l_r + \|\text{sg}(z_e(x)) - z_q(x)\|_2] \tag{12}$$

The overall objective for the stage-II is:

$$\mathcal{L}_{\text{S-II}} = \mathcal{L}^{\mathcal{T}}_{\text{a-e}} + \beta \mathcal{L}^{\mathcal{S}-\mathcal{T}}_{\text{PR}} \tag{13}$$

where β is the hyper-parameter as the same in $\mathcal{L}_{\text{S-I}}$. With the VAE trained using $\mathcal{L}_{\text{S-II}}$, we can obtain utterance-action pairs for system utterances in the domain having no state annotations. However, for utterances having domain-specific intentions, their semantic latent actions are still unclear, which is tackled in Stage 3.

4.4 Stage-III: Domain-specific Actions Learning

We aim to learn semantic latent action for utterances with domain-specific actions at this stage.

Similarity Prediction Network (SPN) We train an utterance-level prediction model, SPN, to predict whether two utterances lead to similar state transitions by taking as input the raw texts of system utterances only. Specifically, SPN gives a similarity score in $[0, 1]$ to an utterance pair:

$$p_{\text{SPN}}(x_i, x_j) = \sigma(r(x_i, x_j)) \tag{14}$$

where $r(\cdot)$ is a scoring function (and we implement it with the same structure as $h(\cdot)$). We use the binary labels a_{ij} indicating whether two utterances x_i and x_j have the same semantic latent action to train the SPN. Specifically, we have $a_{ij} = 1$ if $z_d(x_i) = z_d(x_j)$, and otherwise $a_{ij} = 0$. To facilitate effective knowledge transfer, we obtained such labels from both source and target domains. We consider all pairs of source domain utterances and obtain

$$P^{\mathcal{S}} = \{\langle (x_i, x_j), a_{ij} \rangle \mid x_i, x_j \in U^{\mathcal{S}}\}$$

We also consider pairs of target domain utterances with shared actions: we first get all target domain utterances with aligned actions $U^{\mathcal{T}}_{\text{shared}} = \{x^{\mathcal{T}}_j | x^{\mathcal{T}}_j \in U^{\mathcal{T}}, z_d(x^{\mathcal{T}}_j) \in A^{\mathcal{S}}\}$ where $A^{\mathcal{S}}$ represents the set of shared actions $A^{\mathcal{S}} = \{z_d(x^{\mathcal{S}}_i) \mid x^{\mathcal{S}}_i \in U^{\mathcal{S}}\}$ and then obtain

$$P^{\mathcal{T}} = \{\langle (x_i, x_j), a_{ij} \rangle \mid x_i, x_j \in U^{\mathcal{T}}_{\text{shared}}\}.$$

Using all the collected pairwise training instances $p = \langle (x_i, x_j), a_{ij} \rangle$, we train SPN via the loss

$$\mathcal{L}_{\text{SPN}} = \mathbb{E}_{p \in P^{\mathcal{S}} + P^{\mathcal{T}}}[\text{cross-entropy}(a_{ij}, r(x_i, x_j))]. \tag{15}$$

We then use the trained p_{SPN} to replace state tracking models in both pointwise and pairwise measure. Specifically, we formulate the following pointwise loss

$$\begin{aligned}\mathcal{L}^{\mathcal{T}}_{\text{PT}} &= \mathbb{E}_{x \in U^{\mathcal{T}}}[-\log p_{\text{SPN}}(x^{\mathcal{T}}, \tilde{x}^{\mathcal{T}})] \\ \tilde{x}^{\mathcal{T}} &\sim p_{\mathcal{G}}(z_q(x^{\mathcal{T}}))\end{aligned} \tag{16}$$

which enforces the reconstructed utterances to bring similar dialogue state transitions as the original utterance. We further formulate the pairwise loss as

$$\begin{aligned}\mathcal{L}^{\mathcal{T}-\mathcal{T}}_{\text{PR}} = \sum_{x_i, x_j \in U^{\mathcal{T}}} & -p_{\text{SPN}}(x_i, x_j) \log d(x^{\mathcal{T}}_i, x^{\mathcal{T}}_j) \\ & - (1 - p_{\text{SPN}}(x_i, x_j)) \log(1 - d(x^{\mathcal{T}}_i, x^{\mathcal{T}}_j)) \\ d(x^{\mathcal{T}}_i, x^{\mathcal{T}}_j) &= \sigma(-z_e(x^{\mathcal{T}}_i)^\top z_e(x^{\mathcal{T}}_j)).\end{aligned} \tag{17}$$

Compared to the pairwise loss at stage-I (Eqn. 8) and stage-II (Eqn. 11), the main difference is that we use p_{SPN} to substitute $s_{i,j}$ that relies on trained dialogue state tracking models.

The overall objective function for stage-III is:

$$\mathcal{L}_{\text{S-III}} = \mathcal{L}^{\mathcal{T}}_{\text{a-e}} + \alpha \mathcal{L}^{\mathcal{T}}_{\text{PT}} + \beta \mathcal{L}^{\mathcal{T}-\mathcal{T}}_{\text{PR}} \tag{18}$$

4.5 Conditioned Response Generation

After obtaining semantic latent actions, we train the two components, dialogue planning and surface realization, for conditioned response generation. Specifically, we first train a surface realization model p_r that learns how to translate the semantic latent action into fluent text in context c as

$$\mathcal{L} = \mathbb{E}_x[-\log p_r(x | z_d(x), c)]$$

Then we optimize a dialogue planning model p_l while keeping the parameters of p_r fixed

$$\mathcal{L} = \mathbb{E}_x \mathbb{E}_z[-\log p_r(x|z, c) p_l(z|c)]$$

In this way, the response generation is factorized into $p(x|c) = p(x|z, c)p(z|c)$, where dialogue planning and surface realization are optimized without impinging the other.

5 Experiments

To show the effectiveness of MALA, we consider two experiment settings: multi-domain joint training and cross-domain response generation (Sec. 5.1). We compare against the state-of-the-art on two multi-domain datasets in both settings (Sec. 5.2). We analyze the effectiveness of semantic latent actions and the multi-stage strategy of MALA under different supervision proportion (Sec. 5.3).

Table 2: Multi-Domain Joint Training Results

Model		SMD		MultiWOZ	
		Entity-F1	BLEU	Entity-F1	BLEU
w/o Action	KVRN	48.1	13.2	30.3	11.3
	Mem2seq	62.6	20.5	39.2	14.8
	Sequicity	81.1	21.9	57.7	17.2
w/ Action	LIDM	76.7	17.3	59.4	15.5
	LaRL	80.4	18.2	71.3	14.8
Proposed	MALA-S1	83.8	22.4	74.3	18.7
	MALA-S2	84.7	21.7	76.2	20.0
	MALA-S3	**85.2**	**22.7**	**76.8**	**20.1**

* Note that w/o and w/ Action means whether the baseline considers conditioned generation

Table 3: Cross-Domain Generation Results on SMD

Model		Entity-F1 on target domain			BLEU
		Navigate	Weather	Schedule	
Target Only	Sequicity	31.7	42.6	55.7	16.0
	LaRL	33.2	44.3	57.5	12.3
Fine Tuning	Sequicity	35.9	46.9	59.7	16.8
	LaRL	34.7	45.0	58.6	12.1
Proposed	MALA-S1	38.3	54.8	64.4	19.3
	MALA-S2	39.4	57.0	65.1	18.5
	MALA-S3	**41.8**	**59.4**	**68.1**	**20.2**

5.1 Settings

Datasets We use two multi-domain human-human conversational datasets: (1) SMD dataset (Eric and Manning 2017) contains 2425 dialogues, and has three domains: *calendar*, *weather*, *navigation*; (2) MultiWOZ dataset (Budzianowski et al. 2018) is the largest existing task-oriented corpus spanning over seven domains. It contains in total 8438 dialogues and each dialogue has 13.7 turns in average. We only use five out of seven domains, i.e., *restaurant, hotel, attraction, taxi, train*, since the other two domains contain much less dialogues in training set and do not appear in testing set. This setting is also adopted in the study of dialogue state tracking transferring tasks (Wu et al. 2019). Both datasets contain dialogue states annotations.

We use **Entity-F1** (Eric and Manning 2017) to evaluate dialogue task completion, which computes the F1 score based on comparing entities in delexicalized forms. Compared to inform and success rate originally used on MultiWOZ by Budzianowski et al. (2018), Entity-F1 considers informed and requested entities at the same time and balances the recall and precision. 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 base model.

Multi-domain Joint Training In this setting, we train MALA and other baselines with full training set, i.e., using complete dialogue data and dialogue state annotations. We use the separation of training, validation and testing data as original SMD and MultiWOZ dataset. We compare with the following baselines that do not consider conditioned generation: (1) **KVRN** (Eric and Manning 2017); (2) **Mem2seq** (Madotto, Wu, and Fung 2018); (3) **Sequicity** (Lei et al. 2018); and two baselines that adopt conditioned generation: (4) **LIDM** (Wen et al. 2017a); (5) **LaRL** (Zhao, Xie, and Eskenazi 2019); For a thorough comparison, We include the results of the proposed model after one, two, and all three stages, denoted as **MALA-(S1/S2/S3)**, in both settings.

Cross-domain Response Generation In this setting, we adopt a leave-one-out approach on each dataset. Specifically we use one domain as target domain while the others as source domains. There are three and five possible configurations for SMD and MultiWOZ, respectively. For each configuration, we set that only 1% of dialogues in target domain are available for training, and these dialogues have no state annotations. We compare with Sequicity and LaRL using two types of training schemes in cross-domain response generation. [1] (1) Target only: models are trained only using dialogues in target domain. (2) Fine tuning: model are first trained in the source domains, and we conduct fine-tuning using dialogues in target domain.

5.2 Overall Results

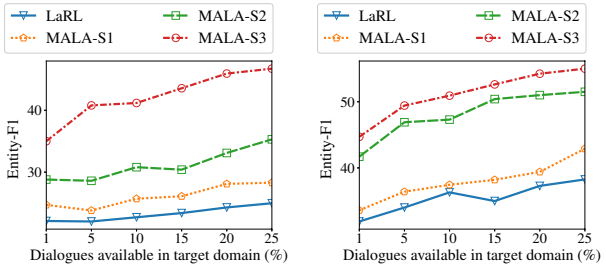
Multi-Domain Joint Training Table 2 shows that our proposed model consistently outperforms other models in the joint training setting. MALA improves dialogue task completion (measured by Entity-F1) while maintaining a high quality of language generation (measured by BLEU). For example, MALA-S3 (76.8) outperforms LaRL (71.3) by 7.71% under Entity-F1 on MultiWOZ, and has the highest BLEU score. Meanwhile, we also find that MALA benefits much from stage-I and stage-II in the joint learning setting. For example, MALA-S1 and MALA-S2 achieve 9.25% and 10.43% improvements over LIDM under Entity-F1 on SMD. This is largely because that, having complete dialogue state annotations, MALA can learn semantic latent actions in each domain at stage-I, and the action alignment at stage-II reduce action space for learning dialogue policy more effectively by finding shared actions across domains. We further find that LIDM and LaRL perform worse than Sequicity on SMD. The reason is that system utterances on SMD have shorter length and various expressions, making it challenging to capture underlying intentions merely based on surface realization. MALA overcomes this challenge by considering dialogue state transitions beyond surface realization in semantic latent action learning.

Cross-Domain Response Generation The results on SMD and MultiWOZ are shown on Tables 3 and 4, respectively. We can see that MALA significantly outperforms the baselines on both datasets. For example, on MultiWOZ, MALA-S3 outperforms LaRL by 47.5% and 55.7% under Entity-F1 using *train* and *hotel* as target domain, respectively. We also find that each stage of MALA is essential in cross-domain generation scenarios. For example, on MultiWOZ using *attraction* as target domain, stage-III and stage-II brings 14.7% and 15.8% improvements compared

[1] We also consider using DAML (Qian and Yu 2019), but the empirical results are worse than those of target only and fine tuning.

Table 4: Cross-Domain Generation Results on MULTIWOZ

MODEL		Hotel		Train		Attraction		Restaurant		Taxi	
		Entity-F1	BLEU	Entity-F1	BLEU	Entity-F1	BLEU	Entity-F1	BLEU	Entity-F1	BLEU
Target Only	Sequicity	16.1	10.7	27.6	16.8	17.4	14.4	19.6	13.9	22.1	15.4
	LaRL	17.8	10.1	30.5	12.9	24.2	11.7	19.9	9.6	28.5	11.7
Fine Tuning	Sequicity	17.3	12.3	27.0	17.6	17.9	15.8	26.0	14.5	22.4	16.9
	LaRL	21.0	9.1	34.7	12.8	24.8	11.8	22.1	10.8	31.9	12.6
Proposed	MALA-S1	23.3	15.5	43.5	18.1	31.5	16.2	24.7	16.5	33.6	18.0
	MALA-S2	26.4	15.8	48.3	18.8	36.5	17.6	28.8	16.6	41.7	18.6
	MALA-S3	**32.7**	**16.7**	**51.2**	**19.4**	**41.9**	**18.1**	**35.0**	**17.3**	**44.7**	**19.0**



(a) Restaurant as target domain (b) Taxi as target domain

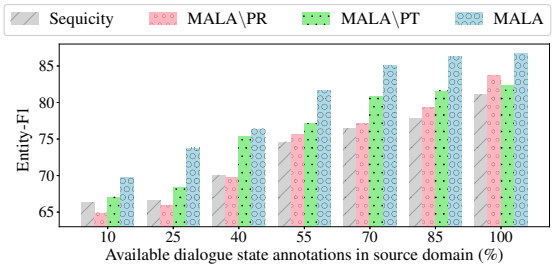
Figure 2: Effects of multiple stages on MULTIWOZ

with its former stage, and MALA-S1 outperforms fine-tuned LaRL by 27.0% under Entity-F1. We further find that the contribution of each stage may vary when using different domains as target, and we will conduct a detailed discussion in the following section. By comparing fine-tuning and target only results of LaRL, we can see latent actions based on lexical similarity cannot well generalize in the cross-domain setting. For example, fine-tuned LaRL only achieves less than 3% over target-only result under Entity-F1 on MultiWOZ using *attraction* as target domain.

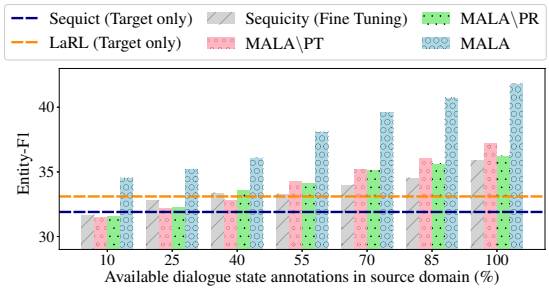
5.3 Discussions

We first study the effects of each stage in MALA in cross-domain dialogue generation. We compare MALA-(S1/S2/S3) with fine-tuned LaRL under different dialogue proportion in target domain. The results are shown in Fig. 2(a) and 2(b). We can see that the performance gain of MALA is largely attributed to stage-III when using *restaurant* as target domain, while attributed to stage-II using *taxi* as target. This is largely because there are many shared actions between *taxi* and *train* domains, many utterance-action pairs learned by action alignment at stage-II already capture the underlying intentions of system utterances. On the other hand, since *restaurant* does not have many shared actions across domains, MALA relies more on the similarity prediction network to provide supervision at stage-III.

Last, we study the effects of semantic latent action in both joint training and cross-domain generation settings. To investigate how pointwise measure $\mathcal{L}_{\text{PT}}$ and pairwise measure $\mathcal{L}_{\text{PR}}$ contribute to capturing utterance intentions, we compare the results of MALA without pointwise loss (MALA\PT), and without pairwise loss (MALA\PR) under varying size of dialogue state annotations. The results



(a) Multi-domain joint training



(b) Cross-domain generation, navigation as target domain

Figure 3: Effects of semantic action learning on SMD

of multi-domain joint training under Entity-F1 on SMD are shown in Fig. 3(a). We can see that both pointwise and pairwise measure are both important. For example, when using 55% of state annotations, encoding pointwise and pairwise measure bring 5.9% and 8.0% improvements, respectively. For cross-domain generation results shown in Fig. 3(b), we can find that these two measures are essential to obtain semantic latent actions in the target domain.

6 Conclusion

We propose multi-stage adaptive latent action learning (MALA) for better conditioned response generation. We develop a novel dialogue state transition measurement for learning semantic latent actions. We demonstrate how to effectively generalize semantic latent actions to the domains having no state annotations. The experimental results confirm that MALA achieves better task completion and language quality compared with the state-of-the-art under both in-domain and cross-domain settings. For future work, we will explore the potential of semantic action learning for zero-state annotations application.

Acknowledgement

We would like to thank Xiaojie Wang for his help. This work is supported by Australian Research Council (ARC) Discovery Project DP180102050, China Scholarship Council (CSC) Grant #201808240008.

References

Budzianowski, P.; Wen, T.-H.; Tseng, B.-H.; Casanueva, I.; Ultes, S.; Ramadan, O.; and Gasic, M. 2018. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. In *EMNLP*, 5016–5026.

Chen, W.; Chen, J.; Qin, P.; Yan, X.; and Wang, W. Y. 2019. Semantically conditioned dialog response generation via hierarchical disentangled self-attention. In *ACL*, 3696–3709.

Chen, Y.-N.; Hakkani-Tür, D.; and He, X. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *ICASSP*.

Eric, M., and Manning, C. D. 2017. Key-value retrieval networks for task-oriented dialogue. *SIGdial*.

Gao, J.; Galley, M.; and Li, L. 2018. Neural approaches to conversational ai. *arXiv preprint arXiv:1809.08267*.

Huang, X.; Qi, J.; Sun, Y.; Zhang, R.; and Zheng, H.-T. 2019. Carl: Aggregated search with context-aware module embedding learning. In *IJCNN*, 101–108. IEEE.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Lei, W.; Jin, X.; Kan, M.-Y.; Ren, Z.; He, X.; and Yin, D. 2018. Sequicity: Simplifying task-oriented dialogue systems with single sequence-to-sequence architectures. In *ACL*.

Madotto, A.; Wu, C.-S.; and Fung, P. 2018. Mem2seq: Effectively incorporating knowledge bases into end-to-end task-oriented dialog systems. In *ACL*, 1468–1478.

Mo, K.; Zhang, Y.; Li, S.; Li, J.; and Yang, Q. 2018. Personalizing a dialogue system with transfer reinforcement learning. In *AAAI*.

Mrkšić, N.; Séaghdha, D.; Thomson, B.; Gašić, M.; Su, P.; Vandyke, D.; Wen, T.; and Young, S. 2015. Multi-domain dialog state tracking using recurrent neural networks. In *ACL*, volume 2, 794–799.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 311–318.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven exploration by self-supervised prediction. In *ICML*, 2778–2787.

Qian, K., and Yu, Z. 2019. Domain adaptive dialog generation via meta learning. In *ACL*, 2639–2649.

Saito, K.; Ushiku, Y.; and Harada, T. 2017. Asymmetric tri-training for unsupervised domain adaptation. In *ICML*.

Shen, T.; Lei, T.; Barzilay, R.; and Jaakkola, T. 2017. Style transfer from non-parallel text by cross-alignment. In *NeurIPS*, 6830–6841.

Sun, X., and Wang, Q. 2019. An internet of things solution for intelligence security management. *ICIS*.

Sun, Y.; Yuan, N. J.; Wang, Y.; Xie, X.; McDonald, K.; and Zhang, R. 2016. Contextual intent tracking for personal assistants. In *SIGKDD*, 273–282. ACM.

Sun, Y.; Yuan, N. J.; Xie, X.; McDonald, K.; and Zhang, R. 2017. Collaborative intent prediction with real-time contextual data. *TOIS* 35(4):30.

van den Oord, A.; Vinyals, O.; and Koray, K. 2017. Neural discrete representation learning. In *NeurIPS*, 6306–6315.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*, 5998–6008.

Wang, X.; Qi, J.; Ramamohanarao, K.; Sun, Y.; Li, B.; and Zhang, R. 2018a. A joint optimization approach for personalized recommendation diversification. In *PAKDD*.

Wang, X.; Zhang, R.; Sun, Y.; and Qi, J. 2018b. Kdgan: knowledge distillation with generative adversarial networks. In *NeurIPS*, 775–786.

Wang, X.; Zhang, R.; Sun, Y.; and Qi, J. 2019. Adversarial distillation for learning with privileged provisions. *TPAMI*.

Wen, T.-H.; Gasic, M.; Mrksic, N.; Su, P.-H.; Vandyke, D.; and Young, S. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *EMNLP*.

Wen, T.-H.; Miao, Y.; Blunsom, P.; and Young, S. 2017a. Latent intention dialogue models. In *ICML*, 3732–3741.

Wen, T.-H.; Vandyke, D.; Mrkšić, N.; Gasic, M.; Barahona, L. M. R.; Su, P.-H.; Ultes, S.; and Young, S. 2017b. A network-based end-to-end trainable task-oriented dialogue system. In *EACL*, 438–449.

Wu, C.-S.; Madotto, A.; Hosseini-Asl, E.; Xiong, C.; Socher, R.; and Fung, P. 2019. Transferable multi-domain state generator for task-oriented dialogue systems. *arXiv preprint arXiv:1905.08743*.

Yang, Z.; Hu, Z.; Dyer, C.; Xing, E. P.; and Berg-Kirkpatrick, T. 2018. Unsupervised text style transfer using language models as discriminators. In *NeurIPS*, 7287–7298.

Yarats, D., and Lewis, M. 2018. Hierarchical text generation and planning for strategic dialogue. In *ICML*, 5587–5595.

Yin, C.; Zhang, R.; Qi, J.; Sun, Y.; and Tan, T. 2018. Context-uncertainty-aware chatbot action selection via parameterized auxiliary reinforcement learning. In *PAKDD*.

Zhang, R.; Guo, J.; Fan, Y.; Lan, Y.; Xu, J.; and Cheng, X. 2018. Learning to control the specificity in neural response generation. In *ACL*, 1108–1117.

Zhao, T., and Eskenazi, M. 2018. Zero-shot dialog generation with cross-domain latent actions. In *SIGdial*, 1–10.

Zhao, T.; Lee, K.; and Eskenazi, M. 2018. Unsupervised discrete sentence representation learning for interpretable neural dialog generation. In *ACL*, 1098–1107.

Zhao, T.; Xie, K.; and Eskenazi, M. 2019. Rethinking action spaces for reinforcement learning in end-to-end dialog agents with latent variable models. In *ACL*, 1208–1218.

Zhong, V.; Xiong, C.; and Socher, R. 2018. Global-locally self-attentive dialogue state tracker. In *ACL*, 1098–1107.